PCT/JP2004/004752

日 本 国 特 許 庁
JAPAN PATENT OFFICE

31.3.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

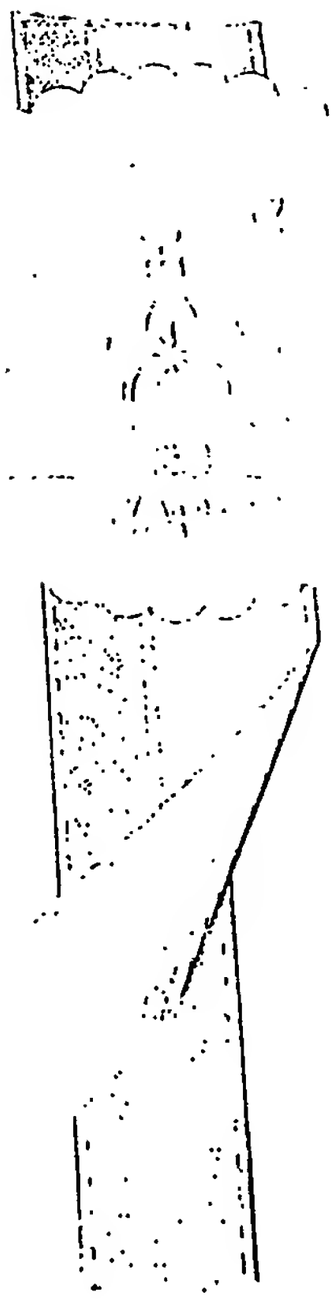
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２００３年 ４月 ９日

出 願 番 号
Application Number: 特願２００３－１０５７６２

[ST. 10/C]: [Ｊ Ｐ ２００３－１０５７６２]

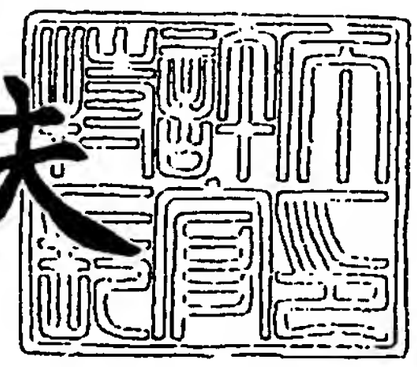
出 願 人
Applicant(s): 財団法人北九州産業学術推進機構



２００４年 ５月１４日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特２００４－３０４０１５０

【書類名】　特許願
【整理番号】　J030129IK0
【あて先】　特許庁長官殿
【国際特許分類】　H03K 19/173
　　G06F　7/00
【発明者】
　【住所又は居所】　福岡県福岡市中央区今川１丁目１０－４３－４０３
　【氏名】　笹尾　勤
【発明者】
　【住所又は居所】　神奈川県足柄下郡箱根町強羅１３２０－１２３１
　【氏名】　井口　幸洋
【特許出願人】
　【識別番号】　502428467
　【氏名又は名称】　笹尾　勤
【代理人】
　【識別番号】　100099508
　【弁理士】
　【氏名又は名称】　加藤　久
　【電話番号】　092-413-5378
【手数料の表示】
　【予納台帳番号】　037590
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

出証特２００４－３０４０１５０

【書類名】　明細書

【発明の名称】　ルックアップテーブル・カスケード論理回路

【特許請求の範囲】

【請求項１】　複数の入力変数を有する目的論理関数を関数分解して得られる複数の分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路であって、

前記各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリと、

前段の前記論理関数メモリの複数個の出力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶する接続メモリと、

前記接続メモリより出力される接続順序に従い、前段の前記論理関数メモリの出力と後段の前記論理関数メモリの入力とを選択的に接続する接続回路と、

最後段の前記論理関数メモリの出力を最前段の前記論理関数メモリに選択的に入力するフィードバック回路と、

を備えていることを特徴とするルックアップテーブル・カスケード論理回路。

【請求項２】　前記フィードバック回路は、

最後段の前記論理関数メモリの複数個の出力のうち、最前段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するフィードバック用接続メモリと、

最後段の前記論理関数メモリの出力を最前段の前記論理関数メモリの入力に選択的に接続するフィードバック用接続回路と、

を備えていることを特徴とする請求項１記載のルックアップテーブル・カスケード論理回路。

【請求項３】　最後段の前記論理関数メモリの複数個の出力を一時的に記憶し、前記フィードバック用接続回路に出力する中間変数バッファを備えていることを特徴とする請求項２記載のルックアップテーブル・カスケード論理回路。

【請求項４】　前記各論理関数メモリ並びに前記各接続メモリ及び前記フィードバック用接続メモリについて、当該論理関数メモリ又は接続メモリ若しくは

フィードバック用接続メモリ内の入出力が可能なページを選択的に切り換えるページ切換手段を備えたことを特徴とする請求項１乃至３の何れか一に記載のルックアップテーブル・カスケード論理回路。

【請求項５】　前記接続回路又はフィードバック用接続回路は、クロスバスイッチ、マルチプレクサ、若しくはシフタ回路により構成されていることを特徴とする請求項１乃至４の何れか一に記載のルックアップテーブル・カスケード論理回路。

【請求項６】　前記接続回路又はフィードバック用接続回路は、前段の前記論理関数メモリ又は最後段の前記論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続することが可能であることを特徴とする請求項１乃至３の何れか一に記載のルックアップテーブル・カスケード論理回路。

【請求項７】　前記各接続回路又はフィードバック用接続回路の前記出力ラインからの出力のうちの何れかを選択し出力する出力段選択回路を備えていることを特徴とする請求項６記載のルックアップテーブル・カスケード論理回路。

【請求項８】　前記出力段選択回路が選択した出力を、
ｗ入力１出力（ｗ≧２）のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした回路を通して出力する出力ライン選択回路を備えていること
を特徴とする請求項７記載のルックアップテーブル・カスケード論理回路。

【請求項９】　前記接続メモリは、前段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するものであり、

前記接続回路は、前記接続メモリより出力される接続順序に従い、前段の前記論理関数メモリの出力と後段の前記論理関数メモリの入力とを選択的に接続するものであり、

前記フィードバック用接続メモリは、最後段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、最前段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するものであり、

前記フィードバック用接続回路は、前記フィードバック用接続メモリより出力される接続順序に従い、最後段の前記論理関数メモリの出力と最前段の前記論理関数メモリの入力とを選択的に接続するものであること
を特徴とする請求項２乃至８の何れか一記載のルックアップテーブル・カスケード論理回路。

【請求項１０】　前記各論理関数メモリにおいては、その入力のうちの少なくとも１つには、前記接続回路を介すことなく前記入力変数を直接入力することを特徴とする請求項９記載のルックアップテーブル・カスケード論理回路。

【請求項１１】　前記各入力変数を記憶する入力変数レジスタと、
前記入力変数レジスタから出力される入力変数のうちの一部を選択し、前記論理関数メモリ又は前記接続回路若しくは前記フィードバック用接続回路に入力する入力選択回路と、
を備えていることを特徴とする請求項１乃至１０の何れか一に記載のルックアップテーブル・カスケード論理回路。

【請求項１２】　前記入力選択回路は、シフタ回路により構成されていることを特徴とする請求項１１に記載のルックアップテーブル・カスケード論理回路。

【請求項１３】　前記入力選択回路は、前記各接続回路若しくはフィードバック用接続回路又は前記各論理関数メモリのそれぞれに対して設けられていることを特徴とする請求項１１又は１２に記載のルックアップテーブル・カスケード論理回路。

【請求項１４】　前記各論理関数メモリは、前段又は最後段の前記論理関数メモリの出力の一部が、前記接続回路又はフィードバック用接続回路を介すことなく後段の前記論理関数メモリの入力の一部に接続されていることを特徴とする請求項１乃至１３の何れか一に記載のルックアップテーブル・カスケード論理回路。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、プログラム可能な論理回路に係り、特に、目的論理関数を関数分解して得られる分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路に関する。

【０００２】

【従来の技術】

近年、論理回路の設計においては、マトリックス状に配列された複数の論理セルの内容と論理セル間の配線の接続とをメモリの書き換えによって変更することが可能なフィールド・プログラマブル・ゲートアレイ（Field Programmable Gate Array。以下、「FPGA」という。）が広く用いられるようになっている。FPGAは、プログラムに従ってソフトウェア的に演算を行うマイクロ・プロセッサ（以下、「MPU」という。）とは異なり、ハードウェア的に演算を行うことから、論理関数の演算実行速度が速いという特徴を有する。

【０００３】

その一方、FPGAは、物理的な配線の引き回しをプログラムにより変更することから、配線遅延を考慮した設計が必要とされる。そのため、論理回路の設計に長時間を要する。また、各論理セル間の配線の引き回し方により配線遅延時間が変わる。そのため、設計時において論理回路の演算時間の予測が困難である。

【０００４】

そこで、上記FPGAの欠点をカバーするプログラム可能な論理回路として、ルックアップテーブル・カスケード論理回路が提案されている（非特許文献１，２参照）。以下、ルックアップテーブル・カスケード論理回路について簡単に説明する。

【０００５】

図２４はルックアップテーブル・カスケード論理回路の原理を表す図である。

ルックアップテーブル・カスケード論理回路において論理関数の演算を行う場合、まず、演算を行う論理関数（目的論理関数）f(X)をp個の分解関数 $\{f_i(X_i) ; i=0, \cdots, p-1\}$ に関数分解する。ここで、$X=(x_0, \cdots, x_{n-1})$は入力変数を表す。また、入力変数の集合を$\{X\}$で表すと、$\{X\}=\{X_0\} \cup \cdots \cup \{X_{p-1}\}$，$\{X_i\} \cap \{X_j\}=\phi$ $(i \neq j;\ i, j \in \{0, \cdots, p-1\})$である。各分解関数$f_i$の出力変数（一般に、ベクト

ル）をY_{i+1}で表す。以下、X,Yの変数の個数を｜X｜，｜Y｜のように表すこととし、特に、｜X｜＝n、｜X_i｜＝n_i、｜Y_i｜＝u_iと表記する。

【０００６】

任意の目的論理関数fは、n_0=s（sは分解関数f_0の入力変数の数）、n_i=s-u_i（0＜i＜p-1）、n_{p-1}=s-u_{p-1}-t（0≦t≦s-2）となるように関数分解することが可能である。そこで、目的論理関数fをp-1個のs入力の分解関数f_j（$j \in \{0, \cdots, p-2\}$）と１個のs-t入力の分解関数f_{p-1}とに分解する。そして、各分解関数f_k（$k \in \{0, \cdots, p-1\}$）を真理値表により表し、これをルックアップテーブルとする。

【０００７】

図２４において、p個のルックアップテーブルLUT_0～LUT_{p-1}は、それぞれ、分解関数 $\{f_i ; i=0, \cdots, p-1\}$ を真理値表により表現したものである。それぞれのルックアップテーブルLUT_kは、s入力u_{k+1}出力の論理関数メモリを用いて実現することができる。このような論理関数メモリを、図２４に示すようにカスケード接続することにより、ルックアップテーブル・カスケード論理回路が実現される。

【０００８】

ルックアップテーブル・カスケード論理回路では、目的論理関数fの入出力変数の個数が大きい場合には、多数のルックアップテーブルをカスケード接続する必要があるため、最適化されたFPGAに比べれば演算速度は遅くなる場合もある。しかし、ルックアップテーブル・カスケード論理回路では、各分解関数の演算について、論理関数メモリを用いてハードウェア的に行うことから、MPUに比べると演算速度を高速化することができる。そして、ルックアップテーブルの段数によって、一義的に演算速度が定まることから、論理回路の設計時に目的論理関数の演算時間を予測することが容易である。また、互いに隣接する論理関数メモリ間でのみ配線を行えばよいことから、論理回路の設計時には配線遅延の影響を考慮する必要がなく、FPGAに比べると論理回路設計が格段に容易になる。

【０００９】

【非特許文献１】

笹尾勤，松浦宗寛，井口幸洋，“多出力関数のカスケード実現と再構

成可能ハードウェアによる実現”，電子情報通信学会FTS研究会，FTS2001-8，pp. 57-64，三重大学(2001-04).

【非特許文献２】

T. Sasao, M. Matsuura, and Y. Iguchi, "A cascade realization of multiple-output function for reconfigurable hardware," International Workshop on Logic and Synthesis (IWLS01), Lake Tahoe, CA, June 12-15, 2001. pp.225-230.

【００１０】

【発明が解決しようとする課題】

しかしながら、実際に図２４に示したようなルックアップテーブル・カスケード論理回路により目的論理関数の演算を行う場合には、目的論理関数を関数分解して得られる分解関数のルックアップテーブル間のレイル数（二つのルックアップテーブル間の線数（中間変数の数））や各ルックアップテーブルの入力変数の個数が目的論理関数により異なる。そのため、種々の目的論理関数をルックアップテーブル・カスケード論理回路で実現しようとした場合、入力変数の入力ラインの数及びレイル数が異なる種々のルックアップテーブル・カスケードが必要となり、実用性に欠ける。

【００１１】

そこで、多様な目的論理関数に対応可能なルックアップテーブル・カスケード論理回路とするために、各論理関数メモリの入力のビット数に余裕をもたせ、あらかじめビット数を大きめにとっておくことが考えられる。しかし、論理関数メモリのメモリ容量は、論理関数メモリの入力のビット数の増加に伴い増大する。そして、使用されないメモリ領域が増大し、無駄が多くなる。また、メモリ容量の増大に伴いメモリのレイアウト面積が増大するため、回路の高集積化が困難となり、回路消費電力も大きくなる。

【００１２】

また、目的論理関数の入力変数が多くなった場合、多数の分解関数に関数分解してルックアップテーブル・カスケードを構成する必要が生じる。しかし、分解関数の数が増加すると、多数の論理関数メモリを直接接続する必要が生じ、回路

面積が大きくなる。従って、ルックアップテーブル・カスケード論理回路の小型化・高集積化が困難となる。

【００１３】

そこで、本発明の目的は、目的論理関数に応じて、各論理関数メモリ間の入力変数の入力ライン数とレイル数とを柔軟に変化させることが可能で、論理関数メモリの入力数、延いては論理関数メモリのメモリ容量を必要最小限に抑えて設計することが可能であり、また、分解関数の数に応じてルックアップテーブル・カスケードの段数を柔軟に変化させることが可能なルックアップテーブル・カスケード論理回路を提供することにある。

【００１４】

【課題を解決するための手段】

本発明に係るルックアップテーブル・カスケード論理回路の第１の構成は、複数の入力変数を有する目的論理関数を関数分解して得られる複数の分解関数を、逐次計算することにより、目的論理関数の演算を行うルックアップテーブル・カスケード論理回路であって、前記各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリと、前段の前記論理関数メモリの複数個の出力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶する接続メモリと、前記接続メモリより出力される接続順序に従い、前段の前記論理関数メモリの出力と後段の前記論理関数メモリの入力とを選択的に接続する接続回路と、最後段の前記論理関数メモリの出力を最前段の前記論理関数メモリに選択的に入力するフィードバック回路とを備えていることを特徴とする。

【００１５】

この構成によれば、目的関数の分解関数の真理値をルックアップテーブルとして論理関数メモリに記憶させておき、目的関数に応じて、各論理関数メモリの入出力の選択及び接続関係を接続メモリに記憶させておく。そして、接続メモリに記憶された選択及び接続関係に従って、接続回路によって前段の論理関数メモリと後段の論理関数メモリの入出力が接続される。これにより、目的関数に適合させて接続関係を自由に変更することができるため、論理回路をプログラマブルに

構成することが可能となる。

【００１６】

また、フィードバック回路によって、最後段の論理関数メモリの出力は、最前段の論理関数メモリに選択的に入力することが可能である。従って、ルックアップテーブル・カスケード論理回路によって演算することが可能な論理関数の大きさの自由度が大きくなる。すなわち、目的関数の入力変数の数が各論理関数メモリの入力端子数の和よりも多い場合であっても、最後段の論理関数メモリの出力を最前段の論理関数メモリにフィードバックさせて、同じ論理関数メモリを重複して使用して分解関数の演算を行わせることができる。従って、分解関数の数に応じてルックアップテーブル・カスケードの段数を柔軟に変化させることが可能となる。そのため、設計自由度が大きくなる。

【００１７】

更に、フィードバック回路を備えたことで、ルックアップテーブル・カスケードの段数を多くすることができるので、各段のルックアップテーブルの大きさは小さくすることができる。従って、各段の論理関数メモリは、メモリ容量の比較的小さいものを使用することができる。そのため、各論理関数メモリの消費電力は低く抑えることが可能である。更に、演算の流れに従って、一部の論理関数メモリのみがデータ読み出しの動作をする。そして、この動作を行うメモリのみが主として電力を消費する。従って、ＭＰＵやＦＰＧＡに比べ、低消費電力で動作させることが可能となる。また、データ読み出しの動作をしていないメモリは、低電力モードにするように構成すれば、更に消費電力の低減が可能となる。

【００１８】

本発明に係るルックアップテーブル・カスケード論理回路の第２の構成は、前記第１の構成において、前記フィードバック回路は、最後段の前記論理関数メモリの複数個の出力のうち、最前段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するフィードバック用接続メモリと、最後段の前記論理関数メモリの出力を最前段の前記論理関数メモリの入力に選択的に接続するフィードバック用接続回路と、を備えていることを特徴とする。

【００１９】

この構成によれば、最後段の論理関数メモリの出力を最前段の論理関数メモリの入力にフィードバックさせる際に、最後段の論理関数メモリから出力される中間変数の選択及び最前段の論理関数メモリとの接続関係を、フィードバック用接続回路とフィードバック用接続メモリによってプログラマブルに設定することができる。従って、フィードバック後に最前段の論理関数メモリで演算を行う分解関数のルックアップテーブルの構成の自由度が高まる。また、フィードバックの際の最前段の論理関数メモリの入出力端子を有効に利用することが可能となるとともに、最前段の論理関数メモリのメモリ容量を効率よく活用することが可能となる。

【００２０】

本発明に係るルックアップテーブル・カスケード論理回路の第３の構成は、前記第２の構成において、最後段の前記論理関数メモリの複数個の出力を一時的に記憶し、前記フィードバック用接続回路に出力する中間変数バッファを備えていることを特徴とする。

【００２１】

この構成により、各論理関数メモリの演算をパイプライン処理によって行った場合に、発振することを防止できる。すなわち、最前段の論理関数メモリから最後段の論理関数メモリへ、順次非同期に演算処理を１ループだけ行った後、最後段の論理関数メモリから出力される中間変数の値を、一旦中間変数バッファに記憶する。そして、中間変数バッファの値が確定した時点で、次のループの演算処理を開始させる。これにより、各論理関数メモリに記憶された分解関数の形によらず、各論理関数メモリの出力が発振により不安定化することが防止される。

【００２２】

本発明に係るルックアップテーブル・カスケード論理回路の第４の構成は、前記第１乃至３の何れか一の構成において、前記各論理関数メモリ並びに前記各接続メモリ及び前記フィードバック用接続メモリについて、当該論理関数メモリ又は接続メモリ若しくはフィードバック用接続メモリ内の入出力が可能なページを選択的に切り換えるページ切換手段を備えたことを特徴とする。

【００２３】

この構成により、ページ切換手段によって論理関数メモリ又は接続メモリ若しくはフィードバック用接続メモリ内の入出力が可能なページを選択的に切り換えることで、１つの論理関数メモリで複数の種類の分解関数の演算を行うことが可能となる。

【００２４】

本発明に係るルックアップテーブル・カスケード論理回路の第５の構成は、前記第１乃至４の何れか一の構成において、前記接続回路又はフィードバック用接続回路は、クロスバスイッチ、マルチプレクサ、若しくはシフタ回路により構成されていることを特徴とする。

【００２５】

このように、接続回路をクロスバスイッチ、マルチプレクサ、若しくはシフタ回路とすることで、簡単な回路により接続回路を実現することができる。

【００２６】

特に、回路の演算速度の高速化が要求される場合には、接続回路としてクロスバスイッチを使用することが好ましい。１本の配線上の切換スイッチの個数が少なく、切換スイッチにおける遅延を極力減らすことが可能だからである。

【００２７】

また、回路の特に小型化が要求される場合には、接続回路としてシフタ回路を使用することが好ましい。接続順序の自由度は少ないが、接続メモリに必要とされる、接続回路の接続切換を制御するための制御線の数を最小限とすることが可能となり、回路のレイアウト面積及び消費電力を小さくできるからである。

【００２８】

本発明に係るルックアップテーブル・カスケード論理回路の第６の構成は、前記第１乃至３の何れか一の構成において、前記接続回路又はフィードバック用接続回路は、前段の前記論理関数メモリ又は最後段の前記論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続することが可能であることを特徴とする。

【００２９】

この構成により、カスケード接続された論理関数メモリうちの任意のものから、その出力を取り出すことができる。従って、目的論理関数の分解関数の段数が少ない場合には、カスケードの途中の論理関数メモリの出力を出力変数として取り出すことで、演算処理を早期に終了させて演算速度を高速化することができる。

【００３０】

本発明に係るルックアップテーブル・カスケード論理回路の第７の構成は、前記第６の構成において、前記各接続回路又はフィードバック用接続回路の前記出力ラインからの出力のうちの何れかを選択し出力する出力段選択回路を備えていることを特徴とする。

【００３１】

この構成により、出力回路により、各接続回路の前記出力ラインからの出力又はフィードバック用接続回路の前記出力ラインからの出力のうち、目的論理関数の演算結果が出力されるものを選択して出力させることができる。

【００３２】

本発明に係るルックアップテーブル・カスケード論理回路の第８の構成は、前記第７の構成において、前記出力段選択回路が選択した出力を、ｗ入力１出力（ｗ≧２）のマルチプレクサが多段にカスケード接続され、各段において、出力を取り出すことを可能とした回路を通して出力する出力ライン選択回路を備えていることを特徴とする。

【００３３】

この構成により、出力段選択回路により選択された段の分解関数の出力変数を出力ライン選択回路によって選択し、出力変数の個数を１／２以下にして出力することが可能となる。すなわち、出力段選択回路により選択された段の分解関数の出力変数の個数が、論理関数メモリの出力可能なラインの数の１／２以下の場合、入力変数のうちの一部のものの値に基づいて、出力ライン選択回路により、更に分解関数の出力変数を選択することができる。すなわち、出力ライン選択回路において更に論理演算を行うことができる。従って、各論理関数メモリに入力する入力変数の個数を削減することができるため、論理関数メモリのメモリ使用

効率が向上する。また、ルックアップテーブル・カスケード論理回路全体で演算が可能な論理関数の入力変数の個数に関する制限数を広げることが可能となる。

【００３４】

本発明に係るルックアップテーブル・カスケード論理回路の第９の構成は、前記第２乃至８の何れか一の構成において、前記接続メモリは、前段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、後段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するものであり、前記接続回路は、前記接続メモリより出力される接続順序に従い、前段の前記論理関数メモリの出力と後段の前記論理関数メモリの入力とを選択的に接続するものであり、前記フィードバック用接続メモリは、最後段の前記論理関数メモリの複数個の出力及び前記入力変数の入力のうち、最前段の前記論理関数メモリの複数個の入力に接続されるものの選択及び接続関係に関する接続情報を記憶するものであり、前記フィードバック用接続回路は、前記フィードバック用接続メモリより出力される接続順序に従い、最後段の前記論理関数メモリの出力と最前段の前記論理関数メモリの入力とを選択的に接続するものであることを特徴とする。

【００３５】

この構成により、前段又は最終段の論理関数メモリから出力される中間変数の個数と、各段の論理関数メモリに入力する入力変数の個数とを、接続メモリ及び接続回路又はフィードバック用接続メモリ及びフィードバック用接続回路を用いて、各分解関数に適合させて変更することが可能となる。すなわち、各段の分解関数のルックアップテーブル間のレイル数（二つのルックアップテーブル間の線数（中間変数の数））が少ないときには、その分だけ入力変数の個数を増やすことができ、逆に各段の分解関数のルックアップテーブル間のレイル数が多いときには入力変数の個数を減らすことができる。従って、各論理関数メモリの限られた数の入力を有効に活用することができ、延いては、各論理関数メモリのメモリ容量を有効に活用することができる。

【００３６】

本発明に係るルックアップテーブル・カスケード論理回路の第１０の構成は、

前記第９の構成において、前記各論理関数メモリにおいては、その入力のうちの少なくとも１つには、前記接続回路を介すことなく前記入力変数を直接入力することを特徴とする。

【００３７】

最初の段の論理関数メモリについては、総ての入力に対して入力変数が入力される。一方、最初の段以外の各論理関数メモリにおいては、その入力の一部には前段の論理関数メモリの出力が入力され、他の入力には、入力変数が入力される。しかしながら、最初の段以外の各論理関数メモリには、最低限１つの入力変数が入力される。従って、最初の段以外の各論理関数メモリの入力のうちの少なくとも１つに、接続回路を介すことなく入力変数を直接入力することにより、接続回路の入力ラインの数を減らすことができる。これにより、接続回路及び接続メモリの出力の配線数を減らすことができるため、回路を小型化することが可能となる。

【００３８】

本発明に係るルックアップテーブル・カスケード論理回路の第１１の構成は、前記第１乃至１０の何れか一の構成において、前記各入力変数を記憶する入力変数レジスタと、前記入力変数レジスタから出力される入力変数のうちの一部を選択し、前記論理関数メモリ又は前記接続回路若しくは前記フィードバック用接続回路に入力する入力選択回路と、を備えていることを特徴とする。

【００３９】

この構成によれば、入力選択回路は、入力変数レジスタに記憶された入力変数のうち、少なくとも実行中の分解関数の演算で使用される入力変数を含む変数を選択して出力する。従って、論理関数メモリ又は接続回路若しくはフィードバック用接続回路に入力される入力変数の個数を減らすことができる。そのため、接続回路及び接続メモリ並びにフィードバック用接続回路及びフィードバック用接続メモリの出力の配線数を減らすことができるため、回路を小型化することが可能となる。

【００４０】

本発明に係るルックアップテーブル・カスケード論理回路の第１２の構成は、

前記第１１の構成において、前記入力選択回路は、シフタ回路により構成されていることを特徴とする。

【００４１】

この構成により、入力変数の選択は、入力選択回路であるシフタ回路と接続回路とにより行われる。すなわち、シフタ回路によって入力レジスタに記憶された入力変数のうち、先頭からｋビットだけシフトさせた入力変数から所定のビット分の入力変数が選択されて接続回路又は論理関数メモリに出力される。接続回路は、入力された入力変数のうち、必要なものを選択して論理関数メモリに入力する。従って、簡易な回路によって入力選択回路を構成することができる。また、入力選択回路における入力変数の選択も、入力変数のシフトビット数を指定するだけでよいため、入力選択回路の制御ラインが少なく、配線面積を減らすことができる。

【００４２】

本発明に係るルックアップテーブル・カスケード論理回路の第１３の構成は、前記第１１又は１２の構成において、前記入力選択回路は、前記各接続回路若しくはフィードバック用接続回路又は前記各論理関数メモリのそれぞれに対して設けられていることを特徴とする。

【００４３】

この構成により、各接続回路及び各論理関数メモリに対して入力する入力変数を、各入力選択回路により個別に選択することができる。従って、各論理関数メモリの演算を同時に行うことが可能であり、各論理関数メモリで並行して演算処理を行うことが可能となる。

【００４４】

本発明に係るルックアップテーブル・カスケード論理回路の第１４の構成は、前記第１乃至１３の何れか一の構成において、前記各論理関数メモリは、前段又は最後段の前記論理関数メモリの出力の一部が、前記接続回路又はフィードバック用接続回路を介すことなく後段の前記論理関数メモリの入力の一部に接続されていることを特徴とする。

【００４５】

最初の段以外の各論理関数メモリにおいては、その入力の一部には前段の論理関数メモリの出力が入力され、他の入力には、入力変数が入力される。しかしながら、最初の段以外の各論理関数メモリには、最低１つは前段の論理関数メモリの出力が入力される。従って、最初の段以外の各論理関数メモリの入力のうちの一部を、接続回路を介すことなく前段の論理関数メモリの出力を直接入力することにより、接続回路の入力ラインの数を減らすことができる。これにより、接続回路及び接続メモリの素子数及び出力の配線数を減らすことができるため、回路を小型化することが可能となる。

【００４６】

【発明の実施の形態】

以下、本発明の一実施形態について、図面を参照しながら説明する。

【００４７】

（実施形態１）

図１は本発明の実施形態１に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【００４８】

本発明の実施形態１に係るルックアップテーブル・カスケード論理回路は、入力変数レジスタ１、入力変数選択回路２－０～２－３、入力選択メモリ３－０～３－３、論理関数メモリ４－０～４－３、接続回路５－０～５－３、接続メモリ６－０～６－３、ページ選択メモリ８－０～８－３、及び演算制御部１０を備えている。尚、本実施形態においては、各回路は共通のクロックによって同期して動作するものとする。

【００４９】

入力変数レジスタ１は、ルックアップテーブル・カスケード論理回路によって演算を行う論理関数（目的論理関数）f(X)の演算に用いられるn個の入力変数X＝（x_1，…，x_n）（但し、n∈自然数）を記憶する。論理関数メモリ４－０～４－３は、目的論理関数fを関数分解して得られる分解関数｛f_i；i∈｛0，1，2，3｝｝の真理値表を、ルックアップテーブルとして記憶する。尚、本実施形態においては、一例として、論理関数メモリ４－ｉ（ｉ＝０，１，２，３）の段数を４段としてい

るが、一般にはこの段数は任意に設定することができる。これらの論理関数メモリ４－０～４－３は、直列に配列され、それぞれ接続回路５－１～５－３を介して直列に接続されている。

【００５０】

また、論理関数メモリ４－３の出力側は、フィードバック用の接続回路５－０を介して論理関数メモリ４－０の入力側に接続されている。

【００５１】

入力変数選択回路２－０～２－３は、入力変数レジスタ１から出力されるn個の入力変数$X=(x_1, \cdots, x_n)$のうちから、それぞれ、各段の論理関数メモリ４－０～４－３に記憶された分解関数$\{f_i; i \in \{0, 1, 2, 3\}\}$の真理値表の入力変数$X_i$（$i \in \{0, 1, 2, 3\}$）を選択し、接続回路５－０～５－３へ出力する。入力選択メモリ３－０～３－３は、それぞれ、入力選択回路２－０～２－３の入力変数の選択に関する情報（以下、「入力選択情報」という。）が記憶されている。入力変数選択回路２－０～２－３は、各入力選択メモリ３－０～３－３から出力される入力選択信号に基づいて、入力変数の選択の切り換えを行う。

【００５２】

各接続回路５－ｉ（$i \in \{1, 2, 3\}$）は、論理関数メモリ４－（ｉ－1）及び入力選択回路２－ｉより入力される、中間変数Y_i及び入力変数X_iを、順序を変えて、後段の論理関数メモリ４－ｉ及び外部出力ライン７－ｉへ接続する。また、接続回路５－０は、論理関数メモリ４－３及び入力選択回路２－０より入力される、中間変数Y_i及び入力変数X_iを、順序を変えて、後段の論理関数メモリ４－０及び外部出力ライン７－０へ接続する。各接続メモリ６－ｉ（$i \in \{0, 1, 2, 3\}$）は、各接続回路５－ｉの接続関係に関する情報（以下、「接続情報」という。）を記憶している。接続回路５－ｉは、接続メモリ６－ｉから出力される接続情報信号に基づき、接続関係の切り換えを行う。

【００５３】

各ページ選択メモリ８－ｉ（$i \in \{0, 1, 2, 3\}$）は、目的関数の演算段数に対応して、論理関数メモリ４－ｉに記憶されたルックアップテーブルのうちの使用するものが格納されているページ番号（以下、「ページ情報」という。）が記憶さ

れている。論理関数メモリ４－ｉは、ページ選択メモリ８－ｉが出力するページ番号に従ってページを設定する。

【００５４】

演算制御部１０は、ルックアップテーブル・カスケード論理回路全体の演算処理の制御を行う。

【００５５】

図２は図１の入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）の構成を表す回路図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

【００５６】

本実施形態の入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）は、図２のようなシフタ回路により構成されている。尚、図２では、説明の都合上、１７入力８出力の入力変数選択回路２－ｉの例を示しているが、入出力の数は、これに限られるものではない。

【００５７】

入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）の入力端子in(00)～in(16)は、入力変数レジスタ１の出力端子に接続されている。従って、入力端子in(00)～in(16)からは入力変数$X=(x_1,\cdots,x_n)$が入力される。また、入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）の出力端子out(00)～out(07)は、接続回路５－ｉの入力側端子の一部に接続されている。

【００５８】

入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）は、入力側から、８ビットシフト回路１１－３、４ビットシフト回路１１－２、２ビットシフト回路１１－１、及び１ビットシフト回路１１－０が直列に接続されている。これにより、入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）は、入力端子in(00)～in(16)の内容を0～16ビットの範囲でシフトさせて、連続する8ビット分を出力端子out(00)～out(07)に出力させることができる。

【００５９】

各シフト回路１１－０～１１－３には、それぞれ１ビットの制御ラインshf0～sh

f3が接続されている。制御ラインshf j（j∈{0，1，2，3}）が0のときは、シフト回路１１－ｉは接続のシフトを行わず、制御ラインshf j（j∈{0，1，2，3}）が1のときは、シフト回路１１－ｉは接続のシフトを行う。尚、入力変数選択回路２－ｉ（ｉ∈{0，1，2，3}）の各制御ラインshf0～shf3は、入力選択メモリ３－ｉの出力に接続されている。すなわち、入力選択メモリ３－０から読み出された４ビットのメモリの内容（すなわち、入力選択情報）が、選択制御信号としてそのまま制御ラインshf0～shf3に出力され、入力選択回路２－ｉ（ｉ∈{0，1，2，3}）における入力変数のシフト量が設定される。

【００６０】

尚、図２では、簡略化のために省略しているが、シフタ回路１１－０～１１－３内の各パストランジスタにより入力変数Xの信号振幅が減衰する。従って、入力変数Xの信号振幅を補償するために、シフタ回路の数段ごとに増幅器（バッファ）を挿入しておくことが必要となる。

【００６１】

図３は図１の論理関数メモリ４－ｉ（ｉ∈{0，1，2，3}．以下、符号４－０～４－３をまとめていう場合には、符号４で表す。）の構成を表すブロック図である。

【００６２】

論理関数メモリ４は、その内部に、mページからなるメモリアレイFM_0～FM_{m-1}を備えている。これらのメモリアレイには、分解関数$f_i(j)$（$i \in \{0, 1, 2, 3\}$，$j \in \{0,\cdots,m-1\}$）の真理値表がルックアップテーブルLUT(j)として格納されている。

【００６３】

尚、ここで、添字「(j)」（$j \in \{0,\cdots,m-1\}$）は、論理関数メモリ４のｊページ目を表す。本実施形態におけるルックアップテーブル・カスケード論理回路では、複数の分解関数$\{f_i(j)\}$を、それぞれの論理関数メモリ４の第ｊページに記憶させておき、目的論理関数に応じて、又はループ回数に応じて、ページを切り換えて、目的論理関数を選択することが可能とされている。

【００６４】

また、論理関数メモリ４－ｉ（ｉ∈{0，1，2，3}）は、ページデコーダ１５及び

アドレスデコーダ１６を備えている。ページデコーダ１５は、ページ選択メモリ8-iから入力されるページ選択番号p（p_k）に基づいて、メモリアレイFM_kのメモリアクセスを可能とする。アドレスデコーダ１６は、接続回路からN_{in}ビットの入力ライン１７を介して入力される変数(X_i, Y_i)（$i \in |0, 1, 2, 3|$．但し、ルックアップテーブル・カスケードの１段目の場合は、$Y_i = \phi$。）に基づき、メモリアレイFM_k内のメモリ内容を読み出すアドレスを選択する。この選択されたアドレスのメモリセルには、分解関数$f_k(j)$の真理値$f_k(j)(X_k, Y_k)$が格納されている。メモリアレイFM_kは、ページデコーダ１５及びアドレスデコーダ１６により、メモリセル内のアドレスが指定されると、当該メモリセル内に記憶されているデータを、中間変数Y_{k+1}＝（$y_{k+1,1}, \cdots, y_{k+1,|Y_{k+1}|}$）として出力ライン１８に出力する。

【００６５】

図４は図１の接続回路5-i（$i \in |0, 1, 2, 3|$．以下、符号5-0～5-3をまとめていう場合には、符号５で表す。）の周辺の回路ブロック図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

【００６６】

本実施形態における接続回路５は、入力ラインi_0～i_{15}からの入力信号を循環的に任意のビット数だけシフトさせて出力ラインo_1～o_{15}に出力するシフタ回路により構成されている。尚、図４においては、説明の都合上、一例として、１６ビット入力１６ビット出力の接続回路を記載しているが、入出力のビット数は、特にこれに限定されない。

【００６７】

接続回路５は、入力ラインi_0～i_{15}の側から、８ビットシフタ回路２０-３、４ビットシフタ回路２０-２、２ビットシフタ回路２０-１、１ビットシフタ回路２０-０が直列に接続された構成とされている。各シフタ回路２０-i（$i \in |0, 1, 2, 3|$）のオン／オフは、接続メモリ６（符号6-0～6-3をまとめて符号６で表す。以下同じ。）の出力に接続された制御ラインs_j（$j \in |0, 1, 2, 3|$）により制御される。

【００６８】

接続回路５－ｋ（ｋ∈｛1，2，3｝）の入力ラインi_0～i_{15}のうち、i_0～i_7は前段の論理関数メモリ４－（ｋ－１）の出力側に接続されており、i_8～i_{15}は入力変数選択回路２－ｋの出力側に接続されている。また、接続回路５－０の入力ラインi_0～i_{15}のうち、i_0～i_7は前段の論理関数メモリ４－３の出力側に接続されており、i_8～i_{15}は入力変数選択回路２－０の出力側に接続されている。

【００６９】

一方、接続回路５－ｋ（ｋ∈｛0，1，2，3｝）の出力ラインo_0～o_{15}のうち、o_0～o_7は、後段の論理関数メモリ４－ｋの入力側に接続されており、o_8～o_{15}は外部出力ライン７－ｋとなっている。

【００７０】

各シフタ回路２０－０～２０－３の接続関係のシフトがないときには、入力ラインi_0～i_{15}は、それぞれ出力ラインo_0～o_{15}に接続される。これにより、前段の論理関数メモリ４－（ｋ－１）の出力である中間変数Y_kは、総て後段の論理関数メモリ４－ｋに入力される。

【００７１】

一方、シフタ回路２０－３のみが接続関係を８ビットシフトさせた場合には、入力ラインi_0～i_7が、それぞれ、出力ラインo_8～o_{15}に接続され、入力ラインi_8～i_{15}が、それぞれ、出力ラインo_0～o_7に接続される。これにより、前段の論理関数メモリ４－（ｋ－１）の出力である中間変数Y_kは、総て外部出力ライン７－ｋに出力される。また、入力変数選択回路２－ｋから出力される入力変数X_kは、総て後段の論理関数メモリ４－ｋに入力される。

【００７２】

また、各シフタ回路２０－０～２０－２により、接続関係をｊビット（１≦ｊ≦７）だけシフトさせた場合には、入力ラインi_{8+j}～i_{15}，i_0～i_{j-1}が、それぞれ、出力ラインo_8～o_{15}に接続され、入力ラインi_j～i_{7+j}が、それぞれ、出力ラインo_0～o_7に接続される。これにより、入力ラインi_8～i_{7+j}に入力されるｊビットの入力変数X_kが、後段の論理関数メモリ４－ｋに入力されるとともに、入力ラインi_j～i_7に入力される８－ｊビットの中間変数Y_kが後段の論理関数メモリ４－ｋに入力される。従って、後段の論理関数メモリ４－ｋに入力する入力変数X_kの変数の個

数|X_k|=n_kと、後段の論理関数メモリ４－ｋに入力する中間変数Y_kの変数の個数|Y$_k$|=u_kとを、$s=n_k+u_k=8$の条件下で自由に変更することが可能となっている。

【００７３】

図５は図１の接続メモリ６の構成を表すブロック図である。

接続メモリ６－ｉ（ｉ∈{0，1，2，3}）の内部には、ｍ個のメモリアレイCM_0～CM_{m-1}が備えられている。各メモリアレイCM_0～CM_{m-1}には、接続回路５－ｉの各シフタ回路２０－（N_c-1）～２０－０（ここで、N_cは接続回路におけるシフタ回路の段数を表し、図４では$N_c=4$である。）のオン／オフ情報を表す接続情報($s_{Nc-1}^{(j)}$，…，$s_0^{(j)}$)（ｊ∈{0，…，m-1}）が記憶されている。尚、ここで、添字「(j)」（ｊ∈{0，…，m-1}）は、ｊページ目に格納された接続情報であることを表す。

【００７４】

これらのメモリアレイCM_0～CM_{m-1}のN_cビットの出力ラインs_0～s_{Nc-1}は、接続回路５－ｉの各シフタ回路２０－０～２０－（N_c－１）の制御ラインとなっている。

【００７５】

また、各接続メモリ６の内部には、ページデコーダ２１が設けられている。ページデコーダ２１は、演算制御部１０から入力されるページ選択番号pkに従って、ｋ番目のメモリアレイCM_kを選択する。ページデコーダ２１により選択されたメモリアレイCM_kは、N_cビットの接続情報($s_{Nc-1}^{(k)}$，…，$s_0^{(k)}$)からなる接続制御信号を接続回路５に出力する。

【００７６】

図６は図１の演算制御部１０の構成を表すブロック図である。

演算制御部１０は、演算ステップレジスタ３１、ステップカウンタ３２、ページカウンタ３４、及び出力コントローラ３５を備えた構成からなる。

【００７７】

演算ステップレジスタ３１には、目的関数を関数分解した後の分解関数の数である演算ステップ数が格納される。ステップカウンタ３２は、現在演算が行われている分解関数の段数をカウントする。このステップカウンタ３２には、演算ステップレジスタ３１に格納された値が設定され、演算処理が１段進行するのに応

じて、内部に格納された値を１づつ減じていくダウンカウンタによって構成されている。ステップカウンタ３２は、そのカウント値ｉが０となったときに、終了信号ENDを出力する。また、ステップカウンタ３２は、外部からリセット信号resetが入力されたとき、カウント値を演算ステップレジスタ３１に格納された値に設定する。

【００７８】

ページカウンタ３４は、入力選択メモリ３、接続メモリ６、及びページ選択メモリ８が出力するデータが格納されている各メモリのページ番号をカウントし、そのカウント値ｋをカウント信号pkとして出力する。ページカウンタ３４は、アップカウンタであり、出力コントローラ３５から出力されるチップ・イネーブル信号CE_3の立ち下がりを検出したときに、カウント値ｋを１だけ増加させる。尚、ページカウンタ３４は、リセット信号resetが入力されたとき、又はステップカウンタ３２から終了信号ENDが入力されたときに、カウント値ｋを０にリセットする。

【００７９】

出力コントローラ３５は、各々の論理関数メモリ４－０～４－３に対して、チップ・イネーブル信号CE_0～CE_3、及びアドレス・ストローブ信号$ADSP_0$～$ADSP_3$を出力することで、論理関数メモリ４－０～４－３の出力制御を行う。

【００８０】

尚、チップ・イネーブル信号CE_iは、ｉ段目の論理関数メモリ４－ｉを活性化する信号であり、チップ・イネーブル信号CE_iが“１”（真値）となった（アサートされた）ときに、論理関数メモリ４－ｉが外部からの入力信号の受け付け及びデータの出力を行うことが可能となる。アドレス・ストローブ信号$ADSP_i$は、論理関数メモリ４－ｉに対して、入力されたアドレスのラッチを制御するための信号である。論理関数メモリ４－ｉは、クロックエッジでアドレス・ストローブ信号$ADSP_i$が“１”（真値）となっている（アサートされている）ときに、入力ラインi(00)～i(07)に入力されているデータを、内部のアドレスレジスタに設定する。また、アドレス・ストローブ信号$ADSP_i$が“１”から“０”（偽値）に遷移すると、内部のアドレスレジスタに設定されているアドレス値をラッチする。尚

、チップ・イネーブル信号CE_iが“０”（偽値）とされた状態では、$ADSP_i$が“１”となっても、入力ラインi(00)～i(07)に入力されているデータはラッチされない。

【００８１】

図７は図６の出力コントローラ３５の内部構成を表すブロック図である。

【００８２】

出力コントローラ３５は、４つのフリップ・フロップ（以下、「FF」という。）４１～４４からなるジョンソンカウンタ５０、４つのAND回路４５～４８、及び１つのOR回路４９によって構成されている。

【００８３】

FF４１～４４のクロック端子Cには、共通のクロック信号clockが入力されている。また、FF４１，４２，４３，４４の出力Q_0，Q_1，Q_2，NOT(Q_3)は、それぞれ、FF４２，４３，４４，４１のデータ入力D_1，D_2，D_3，D_0に接続されている。また、OR回路４９は、リセット信号reset及び終了信号ENDの論理和のリセット信号reset’を出力する。FF４１～４４のリセット端子rstには、リセット信号reset’が入力されている。

【００８４】

このように構成することにより、ジョンソンカウンタ５０の出力（Q_0，Q_1，Q_2，Q_3）の状態は、クロック信号clockが立ち上がるごとに、

（０，０，０，０）

→（１，０，０，０）

→（１，１，０，０）

→（１，１，１，０）

→（１，１，１，１）

→（０，１，１，１）

→（０，０，１，１）

→（０，０，０，１）

→（０，０，０，０）

のようにサイクリックに遷移する。すなわち、各クロックごとに、１ビット右に

シフトして、最上位のビット（Q_3）を反転して、最下位のビット（Q_0）とするという動作を繰り返す。

【００８５】

FF４１，４３の出力Q_0，Q_2が、それぞれ、チップ・イネーブル信号CE_0，CE_1として出力される。また、FF４１，４３の出力NOT(Q_0)，NOT(Q_2)が、それぞれ、チップ・イネーブル信号CE_2，CE_3として出力される。ここで、NOT(Q_i)は、出力Q_iの反転出力である。

【００８６】

また、AND回路４５は、FF４１～４４の出力Q_0，NOT(Q_1)，　NOT(Q_2)，　NOT(Q_3)の論理積をアドレス・ストローブ信号$ADSP_0$として出力する。AND回路４６は、FF４１～４４の出力Q_0，　Q_1，　Q_2，　NOT(Q_3)の論理積をアドレス・ストローブ信号$ADSP_1$として出力する。AND回路４７は、FF４１～４４の出力NOT(Q_0)，　Q_1，　Q_2，　Q_3の論理積をアドレス・ストローブ信号$ADSP_2$として出力する。AND回路４８は、FF４１～４４の出力NOT(Q_0)，NOT(Q_1)，　NOT(Q_2)，　Q_3の論理積をアドレス・ストローブ信号$ADSP_3$として出力する。

【００８７】

尚、論理関数メモリ４-０～４-３は、チップ・イネーブル信号CE_0～CE_3が“０”とされた状態においては、低消費電力モードになるものとする。これにより、演算に使用されない論理関数メモリ４の消費電力が低減され、ルックアップテーブル・カスケード回路全体を、より低消費電力化することができる。

【００８８】

以上のように構成された本実施形態に係るルックアップテーブル・カスケード論理回路において、以下その動作について説明する。

【００８９】

図８は実施形態１に係るルックアップテーブル・カスケード論理回路の動作を表すフローチャート、図９は実施形態１に係るルックアップテーブル・カスケード論理回路の動作時における各信号の変化を表すタイミング図である。

【００９０】

最初に、演算を行おうとする目的論理関数f(X)を関数分解した分解関数$\{f_0$,…

, f_{p-1}｝（1≦p）の真理値表を、ルックアップテーブルとして各論理関数メモリ４－０～４－３に書き込んでおく。ここで、分解関数f_{4k}～f_{4k+3}（k=0, …, [(p-1)/4]）のルックアップテーブルは、論理関数メモリ４－０～４－３の第kページに書き込まれる。尚、同一の論理関数メモリ４において、異なるページに同一のルックアップテーブルが書き込まれることとなる場合には、複数ページに同一のルックアップテーブルを書き込むことはせず、１つのルックアップテーブルのみを書き込むようにする（以下、これを「書込縮約」という）。メモリ容量を節約するためである。また、入力選択メモリ３－０～３－３及び各接続メモリ６－１～６－３には、第０ページから順に上記各分解関数｛$f_0, \cdots, f_{p-1}$｝に対応した入力選択情報及び接続情報を書き込んでおく。また、ページ選択メモリ８－ｉ（ｉ∈｛0, 1, 2, 3｝）には、分解関数f_{4k+i}（k=0, …, [(p-1)/4]）に対応するルックアップテーブルが記憶されている論理関数メモリ４－ｉのページ数p(4k+i)を、第[(k-1)/4]ページに格納する。尚、各メモリへの書き込み機能については、図１では図示していないが、通常のメモリの書き込み方法により行われる。更に、演算制御部１０の演算ステップレジスタ３１には、分解関数の全段数ｐを書き込んでおく。尚、以下では、p>4であるものとして説明する。

【００９１】

以上の書き込みがされた状態において、まず、時間区間T_0において、演算制御部１０は、外部から入力されるリセット信号resetによって、ステップカウンタ３２、ページカウンタ３４のカウント値ｉ，ｋがそれぞれリセットされる（Ｓ１）。このとき、ステップカウンタ３２のカウント値ｉは、演算ステップレジスタ３１内に記憶された値ｐに初期化され、ページカウンタ３４のカウント値ｋは０に初期化される。また、当該リセット信号resetにより、演算制御部１０内の出力コントローラ３５のFF４１～４４がリセットされる（Ｓ２）。

【００９２】

このとき、FF４１～４４からなるジョンソンカウンタ５０は、出力値として、$(Q_0, Q_1, Q_2, Q_3) = (0, 0, 0, 0)$，　$(NOT(Q_0), NOT(Q_1), NOT(Q_2), NOT(Q_3)) = (1, 1, 1, 1)$を出力する。従って、出力コントローラ３５は、各論理関数メモリ４－０，４－１，４－２，４－３に対して、チップ・イネーブル信号CE_0，CE_1，CE_2，CE

3として、それぞれ０，０，１，１を出力する。また、ページカウンタ３４は、カウント信号pkとして０を、入力選択メモリ３-０～３-３、接続メモリ６-０～６-３、及びページ選択メモリ８-０～８-３に対して出力する（Ｓ３）。

【００９３】

これにより、入力選択メモリ３-ｉ（ｉ∈｛0，1，2，3｝）は、第０ページに格納された入力選択情報を、入力変数選択回路２-ｉの制御ラインshf0～shf3に出力する。入力変数選択回路２-ｉは、制御ラインshf0～shf3に入力された入力選択情報に従って、入力端子in(00)～in(16)の一部を出力端子out(00)～out(07)と電気的に接続する。

【００９４】

また、接続メモリ６-ｉ（ｉ∈｛0，1，2，3｝）は、第０ページに格納された接続情報$(s_{Nc-1}(0),\cdots,s_0(0))$を接続回路５-ｉに出力する。これにより、接続回路５-ｉは、各制御ラインより入力される接続情報$(s_3(0),\cdots,s_0(0))$に従って、入力ラインi_8～i_{15}と出力ラインo_0～o_7とを接続する。

【００９５】

更に、ページ選択メモリ８-ｉ（ｉ∈｛0，1，2，3｝）は、第０ページに格納されたページ情報p(0)を論理関数メモリ４-ｉのページデコーダ１３に出力する。これにより、論理関数メモリ４-ｉは、第p(0)ページが選択され得る状態となる。

【００９６】

そして、入力変数レジスタ１は、各入力変数選択回路２-０～２-３に対して、入力変数の出力を開始する（Ｓ４）。これにより、接続回路５-０～５-３には、それぞれ、入力変数X_0，X_1，X_2，X_3が出力される。接続回路５-ｉ（ｉ＝０，１，２，３）に入力された入力変数X_0，X_1，X_2，X_3は、設定された接続順序に従って、論理関数メモリ４-ｉの入力側に出力される。

【００９７】

続いて、時間区間T_1において、次のクロック信号clockの立ち上がりで、ジョンソン・カウンタ５０の出力（Q_0,Q_1,Q_2,Q_3）は、（1,0,0,0）に変化する。これにより、CE_0が“１”となり、論理関数メモリ４-０がアクセス可能な状態となるとともに、CE_2は“０”となる。従って、論理関数メモリ４-０から、データD_0が

出力され始める（Ｓ５）。

【００９８】

また、これとほぼ同時に、AND回路４５の出力である$ADSP_0$が“１”となる。$ADSP_0$が“１”となると、論理関数メモリ４-０は、接続回路５-０から入力される入力変数X_0に基づいて、その内部のアドレスレジスタが制定される（Ｓ６）。そして、論理関数メモリ４-０は、第０ページ内の制定されたアドレスに対応するルックアップテーブルの値（中間変数$Y_{4k+(p-i)}$）を、接続回路５-１に出力し始める。接続回路５-１に出力された中間変数$Y_{4k+(p-i)}$は、設定された接続関係に従って、論理関数メモリ４-１に入力される。また、場合によっては、中間変数$Y_{4k+(p-i)}$の一部は、外部出力ライン７-１へと伝達される。

【００９９】

時間区間T_2において、$ADSP_0$が“１”となってから次のクロック信号clockが立ち上がったときに、ジョンソン・カウンタ５０の出力（Q_0，Q_1，Q_2，Q_3）は、（1,1,0,0）に変化する。これにより、$ADSP_0$は“０”となる（Ｓ７）。これにより、論理関数メモリ４－０の出力データD_0の値が中間変数$Y_{4k+(p-i)}$に確定する。一方、このとき、Q_0=1なので、CE_0は“１”にラッチされている。

【０１００】

次いで、時間区間T_3において、次のクロック信号clockの立ち上がりで、ステップカウンタ３２は、カウント値ｉをデクリメントする（Ｓ８）。また、このとき、ジョンソン・カウンタ５０の値は$(Q_0,Q_1,Q_2,Q_3)=(1,1,1,0)$となる。

【０１０１】

ここで、ステップカウンタ３２のカウント値ｉが０でない場合において（Ｓ９）、出力Q_2の値が１に遷移すると、CE_1が“１”となるとともに、CE_3が“０”となる。それに伴って、論理関数メモリ４-１の出力ラインから出力データD_1が出力され始める（Ｓ１０）。

【０１０２】

また、これとほぼ同時に、AND回路４６の出力である$ADSP_1$が“１”となる（Ｓ１１）。これにより、論理関数メモリ４-１は、接続回路５-１を介して前段の論理関数メモリ４-０から入力される中間変数と新たな入力変数とに基づいてアド

レスが制定される。このアドレスの制定に伴って、論理関数メモリ４－１の出力データD_1が変化し、第pkページ内の制定されたアドレスに対応するルックアップテーブルの値（中間変数$Y_{4k+(p-i)}$）が、接続回路５－２に出力され始める。

【０１０３】

そして、時間区間T_4において、次のクロック信号clockの立ち上がりで、ジョンソン・カウンタ５０の値は$(Q_0, Q_1, Q_2, Q_3)=(1, 1, 1, 1)$となる。これにより、AND回路４６の出力である$ADSP_1$が“０”となり、論理関数メモリ４－１の出力データ$D_1$が、中間変数$Y_{4k+(p-i)}$に確定する（Ｓ１２）。接続回路５－２に出力された中間変数$Y_{4k+(p-i)}$は、設定された接続関係に従って、論理関数メモリ４－２に入力される。また、場合によっては、中間変数$Y_{4k+(p-i)}$の一部は、外部出力ライン７－２へと伝達される。

【０１０４】

次いで、時間区間T_5において、次のクロック信号clockの立ち上がりで、ステップカウンタ３２は、カウント値ｉをデクリメントする（Ｓ１３）。また、このとき、ジョンソン・カウンタ５０の値は$(Q_0, Q_1, Q_2, Q_3)=(0, 1, 1, 1)$となる。

【０１０５】

ここで、ステップカウンタ３２のカウント値ｉが０でない場合（Ｓ１４）、出力Q_0の値が０に遷移すると、CE_2が“１”となるとともに、CE_0は“０”となる。それに伴って、論理関数メモリ４－０の出力が停止するとともに、論理関数メモリ４－２の出力ラインから出力データD_2が出力され始める（Ｓ１５）。

【０１０６】

また、これとほぼ同時に、AND回路４７の出力である$ADSP_2$が“１”となる（Ｓ１６）。これにより、論理関数メモリ４－２は、接続回路５－２を介して前段の論理関数メモリ４－１から入力される中間変数と新たな入力変数とに基づいてアドレスが制定される。このアドレスの制定に伴って、論理関数メモリ４－２の出力データD_2が変化し、第pkページ内の制定されたアドレスに対応するルックアップテーブルの値（中間変数$Y_{4k+(p-i)}$）が、接続回路５－３に出力され始める。

【０１０７】

そして、時間区間T_6において、次のクロック信号clockの立ち上がりで、ジョ

ンソン・カウンタ５０の値は$(Q_0,Q_1,Q_2,Q_3)=(0,0,1,1)$となる。これにより、AND回路４７の出力である$ADSP_2$が“０”となり、論理関数メモリ４－２の出力データ$D_2$が、中間変数$Y_{4k+(p-i)}$に確定する（Ｓ１７）。接続回路５－３に出力された中間変数$Y_{4k+(p-i)}$は、設定された接続関係に従って、論理関数メモリ４－３に入力される。また、場合によっては、中間変数$Y_{4k+(p-i)}$の一部は、外部出力ライン７－３へと伝達される。

【０１０８】

次いで、時間区間T_7において、次のクロック信号clockの立ち上がりで、ステップカウンタ３２は、カウント値ｉをデクリメントする（Ｓ１８）。また、このとき、ジョンソン・カウンタ５０の値は$(Q_0,Q_1,Q_2,Q_3)=(0,0,0,1)$となる。

【０１０９】

ここで、ステップカウンタ３２のカウント値ｉが０でない場合において（Ｓ１９）、出力Q_2の値が０に遷移すると、CE_3が“１”となるとともに、CE_1は“０”となる。それに伴って、論理関数メモリ４－１の出力が停止するとともに、論理関数メモリ４－３の出力ラインから出力データD_3が出力され始める（Ｓ２０）。

【０１１０】

また、これとほぼ同時に、AND回路４８の出力である$ADSP_3$が“１”となる（Ｓ２１）。これにより、論理関数メモリ４－３は、接続回路５－３を介して前段の論理関数メモリ４－２から入力される中間変数と新たな入力変数とに基づいてアドレスが制定される。このアドレスの制定に伴って、論理関数メモリ４－３の出力データD_3が変化し、第pkページ内の制定されたアドレスに対応するルックアップテーブルの値（中間変数$Y_{4k+(p-i)}$）が、接続回路５－０に出力され始める。

【０１１１】

そして、時間区間T_0において、次のクロック信号clockの立ち上がりで、ジョンソン・カウンタ５０の値は$(Q_0,Q_1,Q_2,Q_3)=(0,0,0,0)$となる。これにより、AND回路４８の出力である$ADSP_3$が“０”となり、論理関数メモリ４－３の出力データ$D_3$が、中間変数$Y_{4k+(p-i)}$に確定する（Ｓ２２）。接続回路５－０に出力された中間変数$Y_{4k+(p-i)}$は、設定された接続関係に従って、論理関数メモリ４－０に入力される。また、場合によっては、中間変数$Y_{4k+(p-i)}$の一部は、外部出力ライン

7-0へと伝達される。

【０１１２】

このとき、ページカウンタ３４は、カウント値ｋをインクリメントする（Ｓ２３）。これにより、カウント信号pkの値が１だけ増加し、入力選択メモリ３-０～３-３、接続メモリ６-０～６-３、及びページ選択メモリ８-０～８-３の選択されるページが切り替わる。これに伴って、入力変数選択回路２-０～２-３及び接続回路５-０～５-３の接続関係も切り替わる。尚、ページ選択メモリ８-０～８-３の選択されるページが切り替わった場合であっても、各論理関数メモリは、チップ・イネーブル信号CE_i及びアドレス・ストローブ信号$ADSP_i$が“１”となってアドレスを受け付ける状態とならない限りにおいては、ページが切り替わることはない、

【０１１３】

次いで、時間間隔T_1において、次のクロック信号clockの立ち上がりで、ステップカウンタ３２は、カウント値ｉをデクリメントする（Ｓ２４）。また、このとき、ジョンソン・カウンタ５０の出力Q_0の値が１となる。

ここで、ステップカウンタ３２のカウント値ｉが０でない場合には（Ｓ２５）、再びステップＳ５の動作に戻る。

【０１１４】

上記ステップＳ９，Ｓ１４，Ｓ１９，Ｓ２５において、ステップカウンタ３２のカウント値ｉが０であった場合は、ステップカウンタ３２は終了信号ENDを出力する。これにより、演算制御部１０は演算動作を停止し、演算が終了する。

【０１１５】

以上のように、演算処理は、前段から後段に向かって直列処理により行われる。そして、最後段の論理関数メモリ４-３における演算処理が終了した時点で、まだ総ての分解関数の演算が終了していない場合には、論理関数メモリ４-３から出力される中間変数を再び最前段の論理関数メモリにフィードバックして、演算処理を繰り返す。そして、目的関数ｆの演算結果は、各外部出力ライン7-1～7-4より取り出される。

【０１１６】

尚、本実施形態では接続回路５-１～５-３として、図４のシフタ回路を使用したが、接続回路５-１～５-３としては、図１０に示したクロスバスイッチ５-ｉ’や図１１に示したマルチプレクサ５-ｉ”を使用することも可能である。特に、クロスバスイッチ５-ｉ’は、シフタ回路に比べると切換制御ラインの数が多くなり回路が大きくなるという欠点はあるが、信号が通過するパストランジスタの段数を減らすことができるため、演算速度の高速化が可能となる。

【０１１７】

次に、上記動作をより分かりやすく説明するため、具体的な例を用いてルックアップテーブル・カスケード論理回路の具体的な動作を説明する。

【０１１８】

（例）加算回路

ここでは、簡単な例として、二つの２ｎビット（ｎ≧５）の二進数A=(a_{2n-1}, a_{2n-2}, …, a_1, a_0)，B=(b_{2n-1}, b_{2n-2}, …, b_1, b_0)の加算を行う加算器を、上記ルックアップテーブル・カスケード論理回路によって実行する例を説明する。尚、下位からの桁上げ（キャリー）入力ビットとしてc_{in}も考慮して、以下のような演算を行う加算器を考える。

【数１】

$$\begin{array}{lccccc} & a_{2n-1} & a_{2n-2} & \cdots & a_1 & a_0 \\ & b_{2n-1} & b_{2n-2} & \cdots & b_1 & b_0 \\ +) & & & & & c_{in} \\ \hline c_{out} & S_{2n-1} & S_{2n-2} & \cdots & S_1 & S_0 \end{array}$$

ここで、桁上げ入力ビットc_{in}とは、２ｎビット以上の大きな数の加算を行う場合に、加算器を直列につなげて使用する際に、下位の加算器からの桁上げを表すビットである。桁上げ出力ビットc_{out}も、同様に、上位の加算器に出力する桁上げを表すビットである。

【０１１９】

尚、この例においては、入力変数選択回路２-０～２-３は、図２と同様に、log_2x（log_2xが整数でないときは、log_2xを切り上げた数）段のシフタが直列接続された、ｘ入力８出力のシフタ回路で構成されているものとし、$2n+1 \leqq x$である

ものとする。

【０１２０】

このような二つの２ｎビットの二進数A,Bの加算を行う論理関数をf=f(X)（X=(A,B)）とすると、論理関数fは図１２により表される。この関数は、図１３のように、２ｎ個の分解関数$\{g_0, g_1, \cdots, g_{2n-1}\}$に関数分解することが可能である。ここで、各分解関数は、（数２）又は（数３）のような論理式で表される。

【数２】

$$g_0 = [c_{out}^{(0)}, S_0]$$
$$= [a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in},\ a_0 \oplus b_0 \oplus c_{in}]$$

【数３】

$$g_i = [c_{out}^{(i)}, S_i]$$
$$= [a_i b_i \vee a_i c_{out}^{(i-1)} \vee b_i c_{out}^{(i-1)},\ a_i \oplus b_i \oplus c_{out}^{(i-1)}]$$
$$(i = 1, 2, \cdots, 2n-1)$$

すなわち、各分解関数は、２つの入力変数a_i，b_i及び桁上げを表す中間変数$c_{out}^{(i-1)}$の２を法とする和S_i及び桁上げの中間変数$c_{out}^{(i)}$を出力する論理関数となっている。

【０１２１】

これを、図１のような４つの論理関数メモリ４－０～４－３からなるルックアップテーブル・カスケード論理回路で実現する場合には、図１４に示したように、分解関数$\{g_0, g_1, \cdots, g_{2n-1}\}$を、ｎ個の分解関数組$\{g_0, g_1\}$，$\{g_2, g_3\}$，…，$\{g_{2n-2}, g_{2n-1}\}$に分ける。そして、図１５に示したように、各々の分解関数組を１つにまとめてｎ個の分解関数$f_0, f_1, \cdots, f_{n-1}$で表す。ここで、各分解関数は、（数４）又は（数５）のような論理式で表される３変数入力３変数出力論理関数である。

【数４】

$$
\begin{aligned}
f_0 &= [c_{out}^{(0)}, S_1, S_0] \\
&= [a_1 b_1 \vee (a_1 \vee b_1)(a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in}), \\
&\quad a_1 \oplus b_1 \oplus (a_0 b_0 \vee a_0 c_{in} \vee b_0 c_{in}),\ a_0 \oplus b_0 \oplus c_{in}]
\end{aligned}
$$

【数５】

$$
\begin{aligned}
f_i &= [c_{out}^{(i)}, S_{2i+1}, S_{2i}] \\
&= [a_{2i+1} b_{2i+1} \vee (a_{2i+1} \vee b_{2i+1})(a_{2i} b_{2i} \vee a_{2i} c_{out}^{(i-1)} \vee b_{2i} c_{out}^{(i-1)}), \\
&\quad a_{2i+1} \oplus b_{2i+1} \oplus (a_{2i} b_{2i} \vee a_{2i} c_{out}^{(i-1)} \vee b_{2i} c_{out}^{(i-1)}),\ a_{2i} \oplus b_{2i} \oplus c_{out}^{(i-1)}] \\
&\quad (i = 1, 2, \cdots, n-1)
\end{aligned}
$$

【０１２２】

各々の分解関数f_iの真理値表は、図１６（ａ）のようになる。そこで、図１６（ｂ）の真理値表を、論理関数メモリ４－０の第１ページに記憶させ、図１６（ｃ）の真理値表（ルックアップテーブル）を論理関数メモリ４－０～４－３の第０ページに記憶させる。ここで、それぞれ、S_{2i}，S_{2i+1}，$c_{out}^{(i)}$（i=1, 2, …, n-1）は、それぞれ接続回路５－（i+1）の入力ラインi_2, i_3, i_4に出力されるように論理関数メモリの出力ビットを割り当てる。

【０１２３】

各入力選択メモリ３－０～３－３の第０ページには、それぞれ、０ビットシフト、５ビットシフト、９ビットシフト、１３ビットシフト、第ｋページ（１≦ｋ≦[(n-1)/4]）には、それぞれ、16k+1ビットシフト、16k+5ビットシフト、16k+9ビットシフト、16k+13ビットシフトの情報を記憶させておく。

【０１２４】

接続メモリ６－０の第０ページには、接続情報として、８ビット分シフトさせる情報を記憶させる。接続メモリ６－１～６－３の第０ページには、接続情報として４ビット分シフトさせる情報を記憶させる。また、各接続メモリ６－０～６－３の第ｋページ（１≦ｋ≦[(n-1)/4]）には、接続情報として４ビット分シフトさせる情報を記憶させる。但し、接続メモリ６－α（α＝n mod 4）の第[n/4]ページの接続情報は、最終段の分解関数の出力変数（５ビット）を出力させるた

めの接続情報なので、当該ページには５ビット分シフトさせる情報を記憶させる。尚、４ビット分シフトさせる場合には接続情報として$(s_3, s_2, s_1, s_0)=(0,1,0,0)$を、５ビット分シフトさせる場合には$(s_3, s_2, s_1, s_0)=(0,1,0,1)$を、８ビット分シフトさせる場合には接続情報として$(s_3, s_2, s_1, s_0)=(1,0,0,0)$を接続メモリの各ページに記憶させる。例えば、ｎ＝１３の場合、接続メモリ６の内容は、以下の（表１）のようになる。

【表１】

ページ	接続メモリ 6－0	接続メモリ 6－1	接続メモリ 6－2	接続メモリ 6－3
0	(1, 0, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)
1	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)
2	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)
3	(0, 1, 0, 0)	(0, 1, 0, 1)		

【０１２５】

ページ選択メモリ８－０の第０ページには、ページ情報として１を記憶させておく。また、ページ選択メモリ８－１～８－３の第０ページには、ページ情報として０を記憶させておく。更に、ページ選択メモリ８－０～８－３の第ｋページ（$1 \leqq k \leqq [(n-1)/4]$）には、ページ情報として０を記憶させておく。例えば、ｎ＝１３の場合、接続メモリ６の内容は、以下の（表２）のようになる。

【表２】

ページ	接続メモリ 6－0	接続メモリ 6－1	接続メモリ 6－2	接続メモリ 6－3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0			

【０１２６】

以上のようにプログラムされた状態において、以下、ルックアップテーブル・カスケード論理回路の演算処理を説明する。尚、以下の説明では、簡単のためｎ＝５として説明する。

【０１２７】

まず、入力変数レジスタ１には、変数A,B,c_{in}が入力され記憶される。また、演算ステップレジスタ３１には、ｎ＝５が記憶される。そして、演算制御部１０は、まず、ステップカウンタ３２のカウント値ｉをｎ＝５にリセットし、ページカウンタ３４のカウント値ｋを０にリセットする。更に、ジョンソン・カウンタ５０も０にリセットする。これにより、演算制御部１０のページカウンタ３４は、入力選択メモリ３－０～３－３、接続メモリ６－０～６－３、及びページ選択メモリ８－０～８－３に対してページ選択番号pkとして第０ページを設定する。

【０１２８】

また、入力変数選択回路２－０～２－３は、各々の制御ラインに入力された入力選択情報に従って、入力端子in(00)～in(x-1)のうちの８つを出力端子out(00)～out(07)と電気的に接続する。

【０１２９】

具体的には、入力選択メモリ３－０は、in(00)～in(07)をout(00)～out(07)に接続し、入力選択メモリ３－１は、５ビットシフトさせて、in(05)～in(12)をout(00)～out(07)に接続し、入力選択メモリ３－２は、９ビットシフトさせて、in(09)～in(16)をout(00)～out(07)に接続し、入力選択メモリ３－３は、１３ビットシフトさせて、in(13)～in(20)をout(00)～out(07)に接続する。

【０１３０】

また、接続メモリ６－０～６－３は、各々、第０ページに書き込まれた接続情報$(s_3^{(0)},\cdots,s_0^{(0)})$を接続回路５－０～５－３の各制御ラインに出力する。接続回路５－０～５－３は、各制御ラインより入力される接続情報$(s_3^{(0)},\cdots,s_0^{(0)})$に従って、入力ライン$i_8$～$i_{15}$と出力ライン$o_0$～$o_7$とを接続する。具体的には、接続回路５－０は、８ビットシフトさせて、i_8～i_{15}をそれぞれo_0～o_7に接続し、i_0～i_7をそれぞれo_8～o_{15}に接続する。また、接続メモリ６－１～６－３は、４ビットシフトさせて、i_4～i_{11}をそれぞれo_0～o_7に接続し、i_{12}～i_{15}，i_0～i_3をそれぞれo

8～o15に接続する。

【０１３１】

また、ページ選択メモリ８-０～８-３は、各々、第０ページに書き込まれたページ情報を、論理関数メモリ４-０～４-３のページデコーダ１５に対して出力する。これにより、論理関数メモリ４-０は、１ページがアクセス可能な状態となり、論理関数メモリ４-１～４-３は、１ページがアクセス可能な状態となる。

【０１３２】

次に、演算制御部１０は、入力変数レジスタ１により、入力変数Xの出力を行う。このとき、入力変数選択回路２-１～２-３の入力端子には、それぞれ、（表３）のように入力変数の値が入力される。

【表３】

i(00)	i(01)	i(02)	i(03)	i(04)	i(05)	i(06)	i(07)	i(08)
c_{in}	a_1	a_0	b_1	b_0	a_3	a_2	b_3	b_2
i(09)	i(10)	i(11)	i(12)	i(13)	i(14)	i(15)	i(16)	i(17)
a_5	a_4	b_5	b_4	a_7	a_6	b_7	b_6	a_9
i(18)	i(19)	i(20)						
a_{10}	b_9	b_{10}						

【０１３３】

入力変数選択回路２-０は、(out(00)，out(01)，out(02)，out(03)，out(04)) = (c_{in}，a_1，a_0，b_1，b_0)を出力し、これを接続回路５-０の入力ライン（i_8，i_9，i_{10}，i_{11}，i_{12}）に入力する。尚、出力out(05)～out(07)は使用されないので省略した。接続回路５-０は、８ビットシフトさせて、これらを出力ライン（o_0，o_1，o_2，o_3，o_4）に出力する。このようにして、入力変数(c_{in}，a_1，a_0，b_1，b_0)が論理関数メモリ４-０に入力される。

【０１３４】

次に、演算制御部１０の出力コントローラ３５は、CE_0を“１”に設定して論理関数メモリ４-０のデータ出力を行わせるとともに、$ADSP_0$を“１”に設定して、論理関数メモリ４-０内部のアドレスレジスタに、アドレス(i_0，i_1，i_2，i_3，

i_4, i_5, i_6, i_7)= ($c_{in}, a_1, a_0, b_1, b_0$, -, -, -) を制定する。ここで、「-」はドントケア（０でも１でもよい。）を表す。アドレス制定後に、出力コントローラ３５は、$ADSP_0$を“０”とする。

【０１３５】

アドレス制定後、論理関数メモリ４－０の出力端子には、第１ページに格納されたルックアップテーブルLUT_1のアドレス（$c_{in}, a_1, a_0, b_1, b_0$, -, -, -）に記憶された真理値$f_0(c_{in}, a_1, a_0, b_1, b_0)$の値（-, -, $S_0, S_1, c_{out}^{(0)}$, -, -, -)が現れ、これが接続回路５－１に出力される。すなわち、接続回路５－１の各入力ライン（i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7）には(-, -, $S_0, S_1, c_{out}^{(0)}$, -, -, -)が入力される。

【０１３６】

一方、入力変数選択回路２－１は、(out(00), out(01), out(02), out(03)) = (a_3, a_2, b_3, b_2)を出力し、これを接続回路５－１の入力ライン（i_8, i_9, i_{10}, i_{11}）に入力する。

【０１３７】

接続回路５－１からは、それぞれ、（表４）の値が出力される。

【表４】

o_0	o_1	o_2	o_3	o_4	o_5	o_6	o_7
$c_{out}^{(0)}$	-	-	-	a_3	a_2	b_3	b_2
o_8	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}	o_{15}
-	-	-	-	-	-	S_0	S_1

【０１３８】

接続回路５－１の出力ラインo_8～o_{15}は、外部出力ライン７－１として取り出される。従って、演算結果S_0, S_1が外部出力ライン７－１のうちの２本に出力される。

【０１３９】

次に、演算制御部１０のステップカウンタ３２は、そのカウント値ｉをｉ－１＝４に更新する。そして、演算制御部１０の出力コントローラ３５は、CE_1を“１”に設定して論理関数メモリ４－１のデータ出力を行わせるとともに、$ADSP_1$を

“１”に設定して、論理関数メモリ４－１内部のアドレスレジスタに、アドレス(i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)= ($c_{out}^{(0)}$, -, -, -, a_3, a_2, b_3, b_2) を制定する。アドレス制定後に、出力コントローラ３５は、$ADSP_1$を“０”とする。更に、０段目の論理関数メモリ４－０の出力値は既にアドレスレジスタに取り込んだので、出力コントローラ３５は、CE_0を“０”として、論理関数メモリ４－０の出力を停止させる。

【０１４０】

アドレス制定後、論理関数メモリ４－１の出力端子には、第０ページに格納されたルックアップテーブルLUT_0のアドレス ($c_{out}^{(0)}$, -, -, -, a_3, a_2, b_3, b_2) に記憶された真理値f_1($c_{out}^{(0)}$, a_3, a_2, b_3, b_2)の値(-, -, S_2, S_3, $c_{out}^{(1)}$, - , -, -)が現れ、これが接続回路５－２に出力される。すなわち、接続回路５－２の各入力ライン (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7) には(-, -, S_2, S_3, $c_{out}^{(1)}$, -, -, -)が入力される。

【０１４１】

一方、入力変数選択回路２－２は、(out(00), out(01), out(02), out(03)) = (a_5, a_4, b_5, b_4)を出力し、これを接続回路５－２の入力ライン (i_8, i_9, i_{10}, i_{11}) に入力する。

【０１４２】

接続回路５－２からは、それぞれ、（表５）の値が出力される。

【表５】

o_0	o_1	o_2	o_3	o_4	o_5	o_6	o_7
$c_{out}^{(1)}$	-	-	-	a_5	a_4	b_5	b_4
o_8	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}	o_{15}
-	-	-	-	-	-	S_2	S_3

【０１４３】

接続回路５－２の出力ラインo_8～o_{15}は、外部出力ライン７－２として取り出される。従って、演算結果S_2, S_3が外部出力ライン７－２のうちの２本に出力される。

【０１４４】

以下同様にして、論理関数メモリ４－２、論理関数メモリ４－３においても演算が行われる。そして、論理関数メモリ４－２の演算結果S_4, S_5は外部出力ライン７－３のうちの２本に出力される。また、論理関数メモリ４－３の出力値(－, －, S_6, S_7, $c_{out}^{(3)}$, －, －, －)は、接続回路５－０の入力ラインi_0～i_7にフィードバックされる。

【０１４５】

次に、演算制御部１０のステップカウンタ３２は、そのカウント値ｉをｉ－１＝１に更新する。ここで、論理関数メモリ４－３における演算処理が終了したので、ページカウンタ３４は、そのカウント値ｋを１だけインクリメントし、そのカウント値ｋをカウント信号pkに出力する。

【０１４６】

これに伴い、入力選択メモリ３、接続メモリ６、及びページ選択メモリ８のアクセスページが変更されるため、これらのメモリからの出力も変更される。具体的には、入力選択メモリ３－０から出力されるシフト量の情報が１７ビットシフトに変更される。また、接続メモリ５－０から出力される接続情報が(s_3, s_2, s_1, s_0)＝(0,1,0,0)に変更され、接続メモリ５－１から出力される接続情報が(s_3, s_2, s_1, s_0)＝(0,1,0,1)に変更される。更に、ページ選択メモリ８－０が出力するページ情報p(pk=1)の値が０に変更され、論理関数メモリ４－０は０ページがアクセス可能な状態となる。

【０１４７】

また、これにより、論理関数メモリ４－３の演算結果S_6, S_7は外部出力ライン７－０のうちの２本に出力される。

【０１４８】

そして、上述の場合と同様に、論理関数メモリ４－０において演算処理が行われ、論理関数メモリ４－０の出力値(－, －, S_8, S_9, $c_{out}^{(4)}$, －, －, －)は接続回路５－１の入力ラインi_0～i_7に入力される。接続回路５－１は、これを５ビットシフトして、(S_8, S_9, $c_{out}^{(4)}$)を出力ライン（o_{13}, o_{14}, o_{15}）に出力する。これにより、論理関数メモリ４－０の演算結果(S_8, S_9, $c_{out}^{(4)}$)は外部出力ライン７－１

のうちの３本に出力される。以上で、総ての演算結果が出力ライン７-０～７-３に出力されて演算が終了する。

【０１４９】

尚、上記例においては、各段のルックアップテーブルに入力する入力変数X_0, X_1, …, X_{p-1}は互いに共通の要素を持たない（すなわち、$X_i \cup X_j = \phi$ （$i \neq j$））と仮定したが、本発明におけるルックアップテーブル・カスケード論理回路は、入力変数X_0, X_1, …, X_{p-1}の何れか２つが共通の要素を有するような場合にも使用することが可能である。

【０１５０】

（実施形態２）

図１７は本発明の実施形態２に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【０１５１】

本実施形態においては、接続回路５-０～５-３に入力する入力変数を選択する入力変数選択回路２-０～２-３に加えて、接続回路５-０～５-３を介さずに、論理関数メモリ４-０～４-３に直接入力する入力変数の選択を行う入力変数選択回路６０-０～６０-３及びその入力選択情報を記憶する入力選択メモリ６１-１～６１-３を設けたことを第１の特徴とする。また、前段の論理関数メモリ４-（ｉ-１）（$i \in \{1, 2, 3\}$）から後段の論理関数メモリ４-ｉに入力される中間変数の一部を、接続回路５-ｉを介さずに直接入力するように構成したことを第２の特徴とする。

【０１５２】

このように構成することで、接続回路５-０～５-３の入力ラインの本数を減らすことができる。その結果、接続回路５-０～５-３を小型化することが可能となる。また、接続回路５-０～５-３を図４のようなシフタ回路により構成する場合には、シフタの段数を減らすことが可能であり、演算速度の高速化を図ることができる。

【０１５３】

（実施形態３）

図１８は本発明の実施形態３に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【０１５４】

本実施形態においては、フィードバック用の接続回路の入力側に、最後段の論理関数メモリ４－３の複数個の出力を一時的に記憶し、フィードバック用の接続回路５－０に出力する中間変数バッファ６５を備えていることを特徴とする。

【０１５５】

実施形態１においては、論理関数メモリ４－０～４－３として、クロックに同期して動作するシンクロナスなメモリを使用した。しかし、論理関数メモリ４－０～４－３は、シンクロナスなメモリには限られず、非同期に動作するメモリを使用することも可能である。

【０１５６】

しかしながら、最終段の論理関数メモリ４－３の出力は、最前段の論理関数メモリ４－０の入力にフィードバックされることから、論理関数メモリ４－０～４－３に非同期のメモリを使用した場合、場合によっては発振を起こすおそれがある。

【０１５７】

そこで、本実施形態では、中間変数バッファ６５を備えた構成とし、１演算ループの演算が終了するごとに、論理関数メモリ４－０～４－３の出力値を一旦確定させ、論理関数メモリ４－３が出力する中間変数の値を中間変数バッファ６５に記憶させる。そして、次の演算ループに移り、中間変数バッファ６５に記憶された中間変数を値の論理関数メモリ４－０に入力して演算処理を行うようにする。

【０１５８】

これにより、論理関数メモリ４－０～４－３に非同期なメモリを使用した場合にも、発振することを防止できる。

【０１５９】

（実施形態４）

図１９は本発明の実施形態４に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図、図２０は図１９の出力レジスタ及び出力デコー

ダの構成を表すブロック図、図２１は図２０の記憶素子の構成を表すブロック図である。本実施形態においては、外部出力ライン７-１～７-３に出力された出力変数を一時的に記憶する出力変数レジスタ７１、及び出力変数レジスタ７１が出力変数を取り込むためのロード信号を出力する出力選択デコーダ７２を備えたことを特徴とする。他の構成については、図１と同様であるため、説明は省略する。尚、本実施形態においても、各論理関数メモリ４-０～４-３はクロックにより同期して動作するものとする。

【０１６０】

出力変数レジスタ７１は、３２個の記憶素子M（ｉ，ｊ）（i=1,2,3,4，j=0,1,2,3,4,5,6,7）を備えている。各記憶素子M（ｉ，ｊ）は、データを保持するＤフリップ・フロップ７７と２入力１出力のマルチプレクサ７８とから構成されている。Ｄフリップ・フロップ７７には、共通のクロック信号Clockが入力される。このクロック信号Clockが１のときに、Ｄフリップ・フロップ７７はデータ入力Ｄに入力された値をラッチする。マルチプレクサ７８の出力はＤフリップ・フロップ７７のデータ入力Ｄに接続されている。また、Ｄフリップ・フロップ７７の出力Qは、マルチプレクサ７８の０側の入力Ｄ０に接続されている。マルチプレクサ７８の１側の入力Ｄ１は、外部出力ライン７-ｉのｊ番目のラインに接続されている。マルチプレクサ７８は、ロード信号Loadにより制御され、Load = ０のときに０側の入力を選択し、Load = １のときに１側の入力を選択する。

【０１６１】

出力選択デコーダ７２には、２ビットの出力選択信号ｔが演算制御部１０から入力され、４つの出力選択信号T_1～T_4を出力する。ｔ = (0, 0)のときは、(T1, T2, T3, T4) =(1, 0, 0, 0)、ｔ = (0, 1)のときは、(T1, T2, T3, T4) =(0, 1, 0, 0)、ｔ = (1, 0)のときは、(T1, T2, T3, T4) =(0, 0, 1, 0)、ｔ = (1, 1)のときは、(T1, T2, T3, T4) =(0, 0, 0, 1)が出力される。各出力選択信号T_i（ｉ＝１，２，３，４）は、各記憶素子M（ｉ，ｊ）にロード信号Loadとして入力される。

【０１６２】

演算制御部１０は、演算開始時には、ｔ＝（００）として、論理関数メモリ４

－０が演算結果を出力した後にｔ＝（０１）、論理関数メモリ４－１が演算結果を出力した後にｔ＝（１０）、論理関数メモリ４－２が演算結果を出力した後にｔ＝（１１）のように出力選択信号ｔを切り換えて、演算結果を出力変数レジスタ７１にラッチする。そして、演算が終了した時点で、出力変数レジスタ７１に記憶された出力変数を読み出すことで、演算結果を得ることができる。

【０１６３】

（実施形態５）

図２２は本発明の実施形態５に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【０１６４】

本実施形態においては、出力段選択回路８１、出力ライン選択回路８２、及び入力変数選択回路２－４を備えていることを特徴とする。尚、他の構成については、図１と同様であるため、説明は省略する。

【０１６５】

接続回路５－１～５－３及び論理関数メモリ４－３から出力される変数組は、出力段選択回路８１において何れか一つが選択される。出力段選択回路８１において選択された変数組は、入力変数（F_1, F_2, F_3）及び選択切換変数Selectの値に基づいて、出力ライン選択回路８２によりその総て又は一部が選択され、出力変数として出力される。

【０１６６】

ここで、出力変数の個数が２の冪乗である任意のｎ変数論理関数ｆ($x_0, x_1, \cdots, x_{n-1}$)は、以下の形で表現することが可能である。

【数６】

$$f(x_0, x_1, \cdots, x_{n-3}, x_{n-2}, x_{n-1})$$
$$= x_{n-2}x_{n-1}f_0(x_0, x_1, \cdots, x_{n-3}) \vee x_{n-2}\overline{x}_{n-1}f_1(x_0, x_1, \cdots, x_{n-3})$$
$$\vee \overline{x}_{n-2}x_{n-1}f_2(x_0, x_1, \cdots, x_{n-3}) \vee \overline{x}_{n-2}\overline{x}_{n-1}f_3(x_0, x_1, \cdots, x_{n-3})$$

【０１６７】

従って、上記分解関数f_0～f_3を論理関数メモリ４－０～４－３により実現し、各

分解関数f_0～f_3とx_{n-2}, x_{n-1}等との積項を出力ライン選択回路８２により演算することで、ｎ変数の論理関数の演算を行うことが可能となる。

【０１６８】

出力段選択回路８１は、マルチプレクサにより構成されており、演算制御部１０から出力される出力選択信号ｔに従って、各論理関数メモリ４－０～４－３から出力される出力変数のうちの何れかを選択して出力する。

【０１６９】

ここで、出力選択信号ｔとは、第１の出力選択回路２５が選択する出力ライン束の指定番号（$t \in \{(00),(01),(10),(11)\}$）を表す２ビットの信号である。第１の出力選択回路２５は、ｔ＝(00)のときは接続回路５－１の出力ライン束（図４のo_8～o_{15}）を選択し、ｔ＝(01)のときは接続回路５－２の出力ライン束を選択し、ｔ＝(10)のときは接続回路５－３の出力ライン束を選択し、ｔ＝(11)のときは出力ライン選択回路８２の出力ライン束を選択する。

出力ライン選択回路８２は、入力変数選択回路２－４から入力される入力変数の値に従って、出力段選択回路８１から出力される出力変数のうちの全部又は一部を選択して出力する。尚、入力変数選択回路２－４は、図２に示したものと同様のシフト回路で構成されている。但し、本実施形態では、最終段の論理関数メモリ４－０～４－３の出力が８ビットであるため、入力変数選択回路２－４に使用されるシフト回路の出力は、３ビットとされる。一般には、入力変数選択回路２－４の出力は、最終段の論理関数メモリ４－３の出力の数Ｎに対して、

【数７】

$$\lceil \log_2 N \rceil$$

とされる。以下、入力変数選択回路２－４の出力out(00)～out(02)（図２参照）を、それぞれ、F_0，F_1，F_2で表す。

【０１７０】

図２３は図２２の出力ライン選択回路８２の構成を表すブロック図である。尚、この回路図は、動作原理を説明するために、簡略化して記載している。

【０１７１】

出力ライン選択回路８２は、図２３に示したように、２入力１出力のマルチプレクサ（以下、「MUX」という。）８３～８９を多段にカスケード状に接続し、各段において出力を、MUX９０～MUX９６を通して取り出すことができるように構成されている。尚、図２３では、説明の都合上、出力ラインのビット数N_Cが８ビットの例を示しているが、N_Cは８ビットに限られるものではない。

【０１７２】

図２３において、出力ライン選択回路８２の入力ラインy_0～y_7は、出力段選択回路８１の出力側端子に接続される。この入力ラインy_0～y_7より、出力変数Yの値が入力される。また、出力ライン選択回路８２の出力ライン$f^{(0)}$～$f^{(7)}$からは、入力ラインy_0～y_7のうち選択された出力変数Yの値$(y_0,\cdots,y_{|Y|-1})$が出力される。

【０１７３】

入力ラインy_0,y_1、入力ラインy_2,y_3、入力ラインy_4,y_5、及び入力ラインy_6,y_7は、それぞれMUX８３、MUX８４、MUX８５、及びMUX８６の入力側に接続されている。MUX８３，８４、及びMUX８５，８６の出力は、それぞれMUX８７、及びMUX８８の入力側に接続されている。MUX８７，８８の出力は、MUX８９の入力に接続されている。

【０１７４】

一方、（入力ラインy_0，MUX８３の出力）、（入力ラインy_2，MUX８４の出力）、（入力ラインy_4，MUX８５の出力）、及び（入力ラインy_6，MUX８６の出力）は、それぞれMUX９０、MUX９１、MUX９２、及びMUX９３の入力側に接続されている。また、（入力ラインy_1，MUX８７の出力）、（入力ラインy_3，MUX８９の出力）、及び（入力ラインy_5，MUX８８の出力）は、それぞれMUX９４、MUX９５、及びMUX９６の入力側に接続されている。

【０１７５】

MUX８３～８６は、共通の入力変数F_0により切換制御がされる。すなわち、F_0が“０”のときは、MUX８３，８４，８５，８６は、それぞれ、入力ラインy_0,y_2,y_4,y_6を選択し、F_0が“１”のときは、MUX８３，８４，８５，８６は、それぞれ、入力ラインy_1,y_3,y_5,y_7を選択する。

【０１７６】

MUX８７，８８は、共通の入力変数F_1により切換制御がされる。すなわち、F_1が“０”のときは、MUX８７，８８は、それぞれ、MUX８３，８５を選択し、F_1が“１”のときは、MUX８７，８８は、それぞれ、MUX８４，８６を選択する。

【０１７７】

また、MUX８９は、入力変数F_2により切換制御がされる。すなわち、F_2が“０”のときはMUX８９は、MUX８７を選択し、F_2が“１”のときは、MUX８９は、MUX８８を選択する。

【０１７８】

MUX８３～８６によって、８本の入力ラインy_0～y_7のうちの４本が選択される。MUX８３～８８によって、８本の入力ラインy_0～y_7のうちの２本が選択される。更に、MUX８３～８９によって、８本の入力ラインy_0～y_7のうちの１本が選択される。これは、目的論理関数fの出力変数の個数|f|が４個以下の場合、出力段選択回路８１の出力する中間変数Yに対して、更に入力変数(F_0, F_1, F_2)による論理演算を行うことが可能であることを意味する。従って、論理関数メモリ４－０～４－３により行う演算における入力変数の数を減らすことができる。

【０１７９】

すなわち、例えば、目的論理関数$f(x_0, x_1, \cdots, x_{n-2}, x_{n-1})$を、

【数８】

$$x_{n-1}f'(x_0, x_1, \cdots, x_{n-2}) \vee \overline{x}_{n-1}f'(x_0, x_1, \cdots, x_{n-2})$$

のようにシャノン展開して、関数$f'(x_0, x_1, \cdots, x_{n-2})$を４つの分解関数$f_0(X_0)$，$f_1(X_1, Y_1)$，$f_2(X_2, Y_2)$，$f_3(X_3, Y_3)$（但し、$X_0 \cup X_1 \cup X_2 \cup X_3 = |x_0, x_1, \cdots, x_{n-2}|$）に関数分解し、最終段の分解関数$f_3(X_3, Y_3)$の出力数を４以下となるようにすれば、論理関数メモリ４－０～４－３に入力する入力変数の個数を１個減らすことができる。そして、入力変数x_{n-1}は、入力変数F_0として、出力ライン選択回路８２に入力するようにすればよい。

【０１８０】

同様に、目的論理関数$f(x_0, x_1, \cdots, x_{n-2}, x_{n-1})$を、

【数９】

$$x_{n-1}x_{n-2}f''(x_0,x_1,\cdots,x_{n-3}) \vee x_{n-1}\overline{x}_{n-2}f''(x_0,x_1,\cdots,x_{n-3})$$
$$\vee\overline{x}_{n-1}x_{n-2}f''(x_0,x_1,\cdots,x_{n-3}) \vee \overline{x}_{n-1}\overline{x}_{n-2}f''(x_0,x_1,\cdots,x_{n-3})$$

のようにシャノン展開すれば、論理関数メモリ４－０～４－３に入力する入力変数の個数を２個減らすことができ、

【数１０】

$$x_{n-1}x_{n-2}x_{n-3}f'''(x_0,x_1,\cdots,x_{n-4}) \vee x_{n-1}x_{n-2}\overline{x}_{n-3}f'''(x_0,x_1,\cdots,x_{n-4})$$
$$\vee x_{n-1}\overline{x}_{n-2}x_{n-3}f'''(x_0,x_1,\cdots,x_{n-4}) \vee x_{n-1}\overline{x}_{n-2}\overline{x}_{n-3}f'''(x_0,x_1,\cdots,x_{n-4})$$
$$\vee\overline{x}_{n-1}x_{n-2}x_{n-3}f'''(x_0,x_1,\cdots,x_{n-4}) \vee \overline{x}_{n-1}x_{n-2}\overline{x}_{n-3}f'''(x_0,x_1,\cdots,x_{n-4})$$
$$\vee\overline{x}_{n-1}\overline{x}_{n-2}x_{n-3}f'''(x_0,x_1,\cdots,x_{n-4}) \vee \overline{x}_{n-1}\overline{x}_{n-2}\overline{x}_{n-3}f'''(x_0,x_1,\cdots,x_{n-4})$$

のようにシャノン展開すれば、論理関数メモリ４－０～４－３に入力する入力変数の個数を３個減らすことができる。

【０１８１】

尚、MUX８３～MUX８９により選択された８ビット、４ビット、２ビット、又は１ビットの出力は、共通の８本の出力ラインによって出力するのが好ましい。ルックアップテーブル・カスケード論理回路をＬＳＩにより構成した場合、出力ライン数（ピン数）が少ないほどパッケージ面積を縮小できるからである。そこで、図２３では、MUX９０～９６によって、上記各本数の出力ラインが選択されたときの出力を、共通の出力ライン$f^{(0)}$～$f^{(7)}$を通して出力されるように構成されている。MUX９０～９６は、共通の選択切換変数Selectによって切換制御がされる。尚、選択切換変数Selectの設定値は、出力変数の個数|f(X)|によって、あらかじめ選択切換メモリ９７に記憶される。そして、演算制御部１０の制御によって、Selectに設定される。

【０１８２】

MUX９０は、Selectが“０”のとき入力ラインy_0を選択し、Selectが“１”のときMUX８３の出力を選択し、選択したラインの信号値を出力ライン$f^{(0)}$に出力する。MUX９１は、Selectが“０”のとき入力ラインy_2を選択し、Selectが“１”のときMUX８４の出力を選択し、選択したラインの信号値を出力ライン$f^{(2)}$に出力する。MUX９２は、Selectが“０”のとき入力ラインy_4を選択し、Selectが

“１”のときMUX８５の出力を選択し、選択したラインの信号値を出力ラインf(4)に出力する。MUX９３は、Selectが“０”のとき入力ラインy_6を選択し、Selectが“１”のときMUX８６の出力を選択し、選択したラインの信号値を出力ラインf(6)に出力する。

【０１８３】

MUX９４は、Selectが“０”のとき入力ラインy_1を選択し、Selectが“１”のときMUX８７の出力を選択し、選択したラインの信号値を出力ラインf(1)に出力する。MUX９５は、Selectが“０”のとき入力ラインy_3を選択し、Selectが“１”のときMUX８９の出力を選択し、選択したラインの信号値を出力ラインf(3)に出力する。MUX９６は、Selectが“０”のとき入力ラインy_5を選択し、Selectが“１”のときMUX８８の出力を選択し、選択したラインの信号値を出力ラインf(5)に出力する。また、出力ラインf(7)は、入力ラインy_7に直結されている。

【０１８４】

目的論理関数fの演算結果f(X)のビット数（出力の本数）が８のときは、Selectを“０”とする。これにより、出力ラインf(0)～f(7)には、入力ラインy_0～y_7に入力される出力変数$Y=(y_0,\cdots,y_7)$の値が出力される。出力の本数が４のときは、Selectを“１”に設定し、入力変数F_0に“０”又は“１”を入力する。これにより、入力ラインy_0～y_7のうち$F_0=0$のときは（y_0，y_2，y_4，y_6）の４本が、$F0=1$のときは（y_1，y_3，y_5，y_7）の４本が、出力ラインf(0)，f(2)，f(4)，f(6)に出力される。出力の本数が２のときは、Selectを“１”とし、入力変数F_0、F_1に“０”又は“１”を設定する。これにより、$(F_0, F_1)=(0, 0)$のときは、出力ラインf(1)，f(5)には入力ラインy_0～y_7のうち２本に入力される出力変数Y＝（y_0，y_4）の値が出力される。$(F_0, F_1)=(0, 0), (0, 1), (1, 1)$のときは出力には$(y_1, y_5)$，$(y_2, y_6)$，$(y_3, y_7)$の値がそれぞれ出力される。出力の本数が１のときは、Selectを“１”とし、入力変数F_0，F_1，F_2に“０”又は“１”を設定する。これにより、出力ラインf(3)には、入力ラインy_0～y_7のうち１本にF_0，F_1，F_2で指定した入力が出力される。

【０１８５】

このように出力ライン選択回路８２を設けることにより、各論理関数メモリ４

-０～４-３より出力される出力変数の個数が、全出力ラインの本数の１／２以下の場合には、出力ライン選択回路８２を用いて更に論理演算を行うことが可能である。これによって、論理関数メモリ４-０～４-３において演算を行う分解関数の入力変数の総数を１個以上減らすことが可能となる。従って、ルックアップテーブル・カスケード論理回路全体で許容される入力変数の数を増やすことが可能となる。

【０１８６】

尚、図２３では、出力ライン選択回路８２内のマルチプレクサは２入力１出力のものを使用しているが、一般に、ｗ入力１出力（ｗ≧２）のマルチプレクサを使用することが可能である。

【０１８７】

【発明の効果】

以上のように本発明の第１の構成によれば、目的関数に適合させて接続関係を自由に変更することができるため、論理回路をプログラマブルに構成することが可能となる。そして、目的関数の入力変数の数が各論理関数メモリの入力端子数の和よりも多い場合であっても、フィードバック回路によって最後段の論理関数メモリの出力を最前段の論理関数メモリにフィードバックさせて、同じ論理関数メモリを重複して使用して分解関数の演算を行わせることが可能となる。すなわち、演算が可能な目的関数の入力変数の個数の自由度が非常に大きく、設計自由度の大きいルックアップテーブル・カスケード論理回路を提供することができる。

【０１８８】

本発明の第２の構成によれば、最後段の論理関数メモリから出力される中間変数の選択及び最前段の論理関数メモリとの接続関係を、フィードバック用接続回路とフィードバック用接続メモリによってプログラマブルに設定することができるため、フィードバック後に最前段の論理関数メモリで演算を行う分解関数のルックアップテーブルの構成の自由度が高いルックアップテーブル・カスケード論理回路を提供することができる。また、フィードバックの際の最前段の論理関数メモリの入出力端子を有効に利用することが可能となるとともに、最前段の論理

関数メモリのメモリ容量を効率よく活用することが可能となるため、演算が可能な目的関数の自由度の高いルックアップテーブル・カスケード論理回路を小容量の論理関数メモリで実現することが可能となる。

【０１８９】

本発明の第３の構成によれば、ルックアップテーブル・カスケード論理回路をパイプライン処理により非同期的に動作させる場合であっても、各論理関数メモリに記憶された分解関数の形によらず、各論理関数メモリの出力が発振により不安定化することを防止することが可能となる。

【０１９０】

本発明の第４の構成によれば、ページ切換手段によって論理関数メモリ又は接続メモリ若しくはフィードバック用接続メモリ内の入出力が可能なページを選択的に切り換えることで、１つの論理関数メモリで複数の種類の分解関数の演算を行うことが可能となる。従って、複数の論理関数の演算を１個のルックアップテーブル・カスケード論理回路によって行うことが可能となる。また、最後段の論理関数メモリの出力を最前段の論理関数メモリの入力にフィードバックさせて使用する場合、各演算のループによって論理関数メモリ若しくは接続メモリ又はフィードバック用接続メモリのページを切り換えて使用することで、ルックアップテーブル・カスケード論理回路によって演算することが可能な論理関数の自由度を広げることができる。

【０１９１】

本発明の第５の構成によれば、簡単な回路により接続回路を実現することが可能となる。

【０１９２】

本発明の第６の構成によれば、接続回路が、前段の論理関数メモリの出力を、論理関数の演算結果を出力する出力ラインに接続可能としたことで、目的論理関数の分解関数の段数が少ない場合には、カスケードの途中の論理関数メモリの出力を出力変数として取り出し、演算速度を高速化することが可能となる。

【０１９３】

本発明の第７の構成によれば、各論理関数メモリから出力される出力を、出力

段選択回路により自由に選択して出力させることができるので、外部への出力ラインの本数を減らすことが可能となる。従って、集積回路で構成する場合は、出力変数を出力するための集積回路のピン数を削減することが可能となり、小型化することができる。

【０１９４】

本発明の第８の構成によれば、第１の選択された段の出力変数を、第１の選択された段の分解関数の出力変数の個数が、第１の選択された段の論理関数メモリの出力数の１／２以下の場合、入力変数のうちの一部のものの値に基づいて、第１の選択された段の分解関数の出力変数を選択することが可能となり、出力ライン選択回路において更に論理演算を行うことができる。従って、各論理関数メモリに入力する入力変数の個数を削減することができる。そのため、論理関数メモリのメモリ使用効率が向上する。また、ルックアップテーブル・カスケード論理回路全体で演算が可能な論理関数の入力変数の個数に関する制限数を広げることが可能となる。

【０１９５】

本発明の第９の構成によれば、前段又は最終段の論理関数メモリから出力される中間変数の個数と、各段の論理関数メモリに入力する入力変数の個数とを、接続メモリ及び接続回路又はフィードバック用接続メモリ及びフィードバック用接続回路を用いて、各分解関数に適合させて変更することが可能となるため、各論理関数メモリのメモリ容量を有効に活用することができる。

【０１９６】

本発明の第１０の構成によれば、最初の段以外の各論理関数メモリの入力のうちの少なくとも１つに、接続回路を介すことなく入力変数を直接入力することにより、接続回路の入力ラインの数を減らすことができる。そのため、接続回路及び接続メモリの出力の配線数を減らすことができ、回路の高集積化及び省電力化が可能となる。

【０１９７】

本発明の第１１の構成によれば、論理関数メモリ又は接続回路若しくはフィードバック用接続回路に入力される入力変数の個数を減らすことができるため、接

続回路及び接続メモリ並びにフィードバック用接続回路及びフィードバック用接続メモリの出力の配線数を減らすことができ、回路を高集積化及び省電力化が可能となる。

【０１９８】

本発明の第１２の構成によれば、簡易な回路によって入力選択回路を構成することができ、また、入力選択回路における入力変数の選択も入力変数のシフトビット数を指定するだけでよいため、入力選択回路の制御ラインが少なくなる。従って、スイッチング素子の数や配線面積を減らすことができるため、高集積化及び省電力化を行うことが可能となる。

【０１９９】

本発明の第１３の構成によれば、各論理関数メモリの演算を同時に行うことが可能であり、各論理関数メモリで並行して演算処理を行うことが可能となるため、演算速度を向上させることが可能となる。

【０２００】

本発明の第１４の構成によれば、最初の段以外の各論理関数メモリの入力のうちの一部を、接続回路を介すことなく前段の論理関数メモリの出力を直接入力することにより、接続回路の入力ラインの数を減らすことができる。これにより、接続回路及び接続メモリの素子数及び出力の配線数を減らすことができるため、回路を高集積化・省電力化することが可能となる。

【図面の簡単な説明】

【図１】　本発明の実施形態１に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図２】　図１の入力変数選択回路２－ｉ（ｉ∈｛0，１，２，３｝）の構成を表す回路図である。

【図３】　図１の論理関数メモリ４の構成を表すブロック図である。

【図４】　図１の接続回路５の周辺の回路ブロック図である。

【図５】　図１の接続メモリ６の構成を表すブロック図である。

【図６】　図１の演算制御部１０の構成を表すブロック図である。

【図７】　図６の出力コントローラ３５の内部構成を表すブロック図である

。

【図８】　実施形態１に係るルックアップテーブル・カスケード論理回路の動作を表すフローチャートである。

【図９】　実施形態１に係るルックアップテーブル・カスケード論理回路の動作時における各信号の変化を表すタイミング図である。

【図１０】　図１の接続回路をクロスバスイッチにより構成した例を表す図である。

【図１１】　図１の接続回路をマルチプレクサにより構成した例を表す図である。

【図１２】　２ｎビットの二進数A,Bの加算を行う論理関数fを表す図である。

【図１３】　２ｎビットの二進数A,Bの加算を行う論理関数fを２ｎ個の分解関数$\{g_0, g_1, \cdots, g_{2n-1}\}$に関数分解した図である。

【図１４】　２ｎビットの二進数A,Bの加算を行う論理関数fを２ｎ個の分解関数$\{g_0, g_1, \cdots, g_{2n-1}\}$に関数分解した図である。

【図１５】　２ｎビットの二進数A,Bの加算を行う論理関数fをｎ個の分解関数$\{f_0, f_1, \cdots, f_{n-1}\}$に関数分解した図である。

【図１６】　分解関数f_iの真理値表である。

【図１７】　本発明の実施形態２に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図１８】　本発明の実施形態３に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図１９】　本発明の実施形態４に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図２０】　図１９の出力レジスタ及び出力デコーダの構成を表すブロック図である。

【図２１】　図２０の記憶素子の構成を表すブロック図である。

【図２２】　本発明の実施形態５に係るルックアップテーブル・カスケード論理回路の全体構成を表すブロック図である。

【図２３】　図２２の出力ライン選択回路８２の構成を表すブロック図である。

【図２４】　ルックアップテーブル・カスケード論理回路の原理を表す図である。

【符号の説明】

１　入力変数レジスタ

２-０～２-４，６０-０～６０-３　入力変数選択回路

３-０～３-４，６１-１～６１-３　入力選択メモリ

４-０～４-３　論理関数メモリ

５-０～５-３　接続回路

６-０～６-３　接続メモリ

８-０～８-３　ページ選択メモリ

１０　演算制御部

１１-０，２０-０　１ビットシフト回路

１１-１，２０-１　２ビットシフト回路

１１-２，２０-２　４ビットシフト回路

１１-３，２０-３　８ビットシフト回路

FM_0～FM_{m-1}　メモリアレイ

１５，２１　ページデコーダ

１６　アドレスデコーダ

１７　入力ライン

１８　出力ライン

３１　演算ステップレジスタ

３２　ステップカウンタ

３３　ループカウンタ

３４　ページカウンタ

３５　出力コントローラ

４１～４４　フリップ・フロップ（FF）

４５～４８　AND回路

４９　OR回路

６５　中間変数バッファ

７１　出力変数レジスタ

７２　出力選択デコーダ

７７　Ｄフリップ・フロップ

７８，８３～９６　マルチプレクサ（MUX）

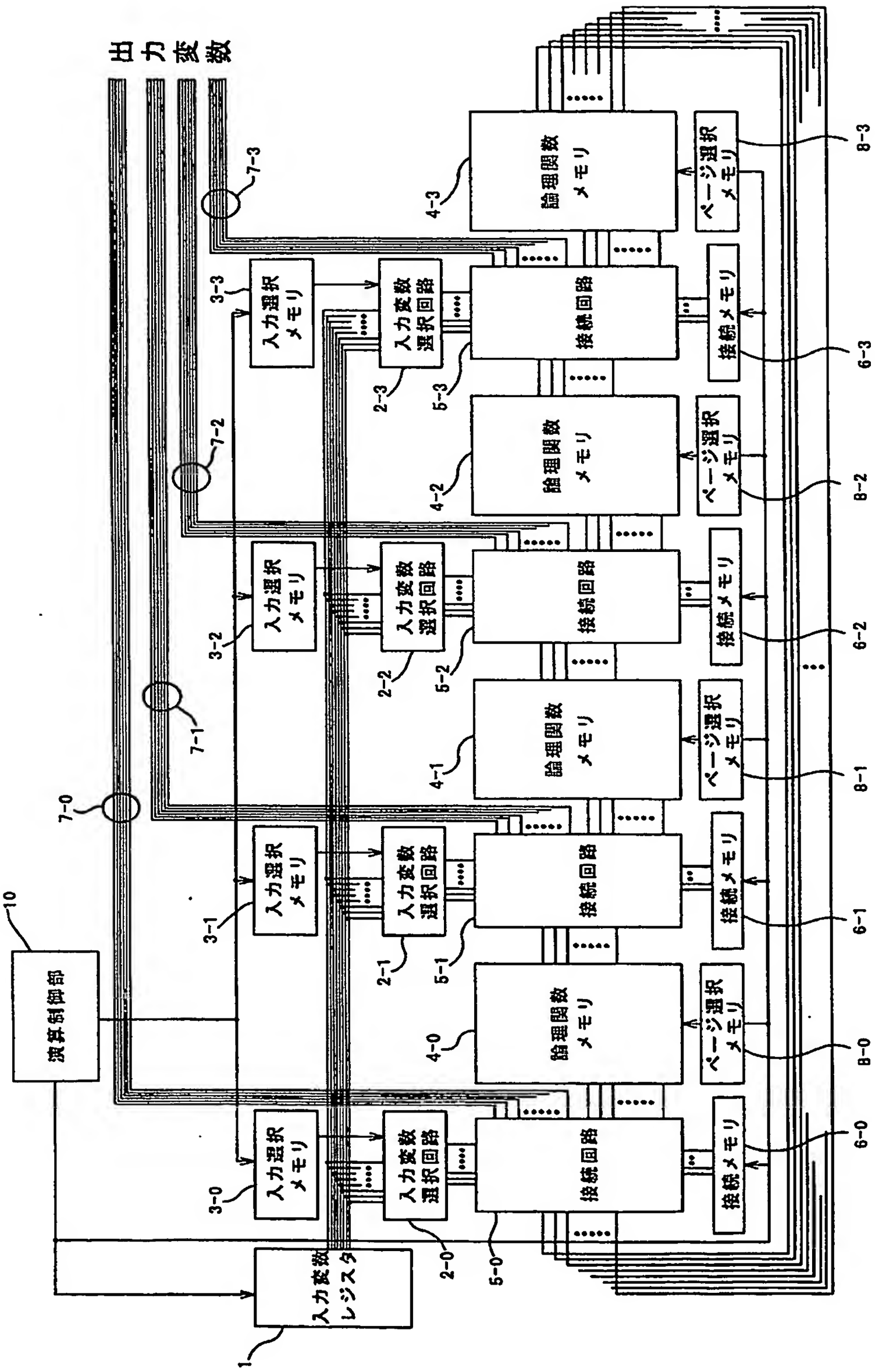
８１　出力段選択回路

８２　出力ライン選択回路

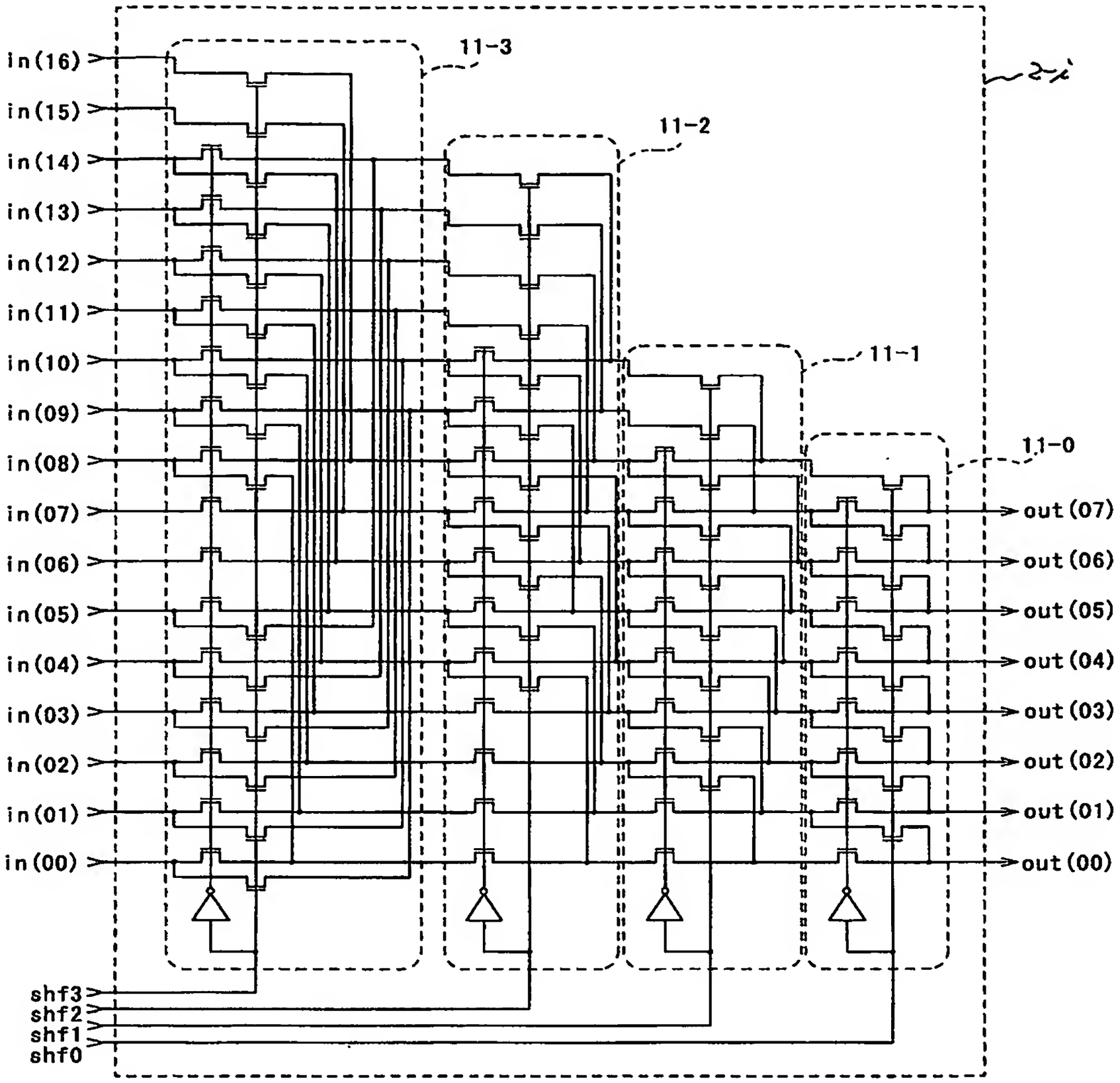
９７　選択切換メモリ

【書類名】　図面

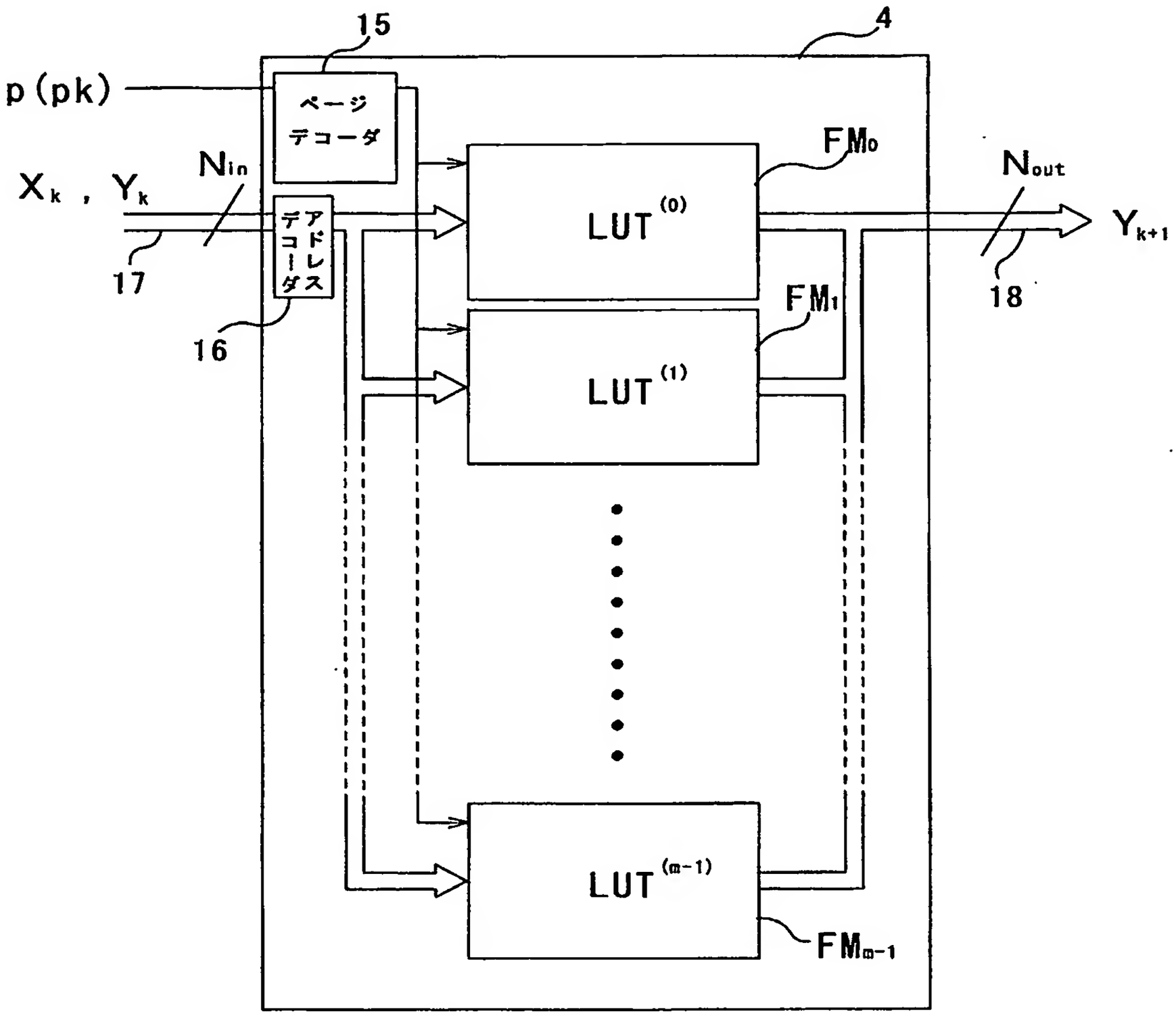
【図１】


出力数
7-3
4-3
論理関数メモリ
ページ選択メモリ
8-3
3-3
入力選択メモリ
入力変数選択回路
2-3
5-3
接続回路
接続メモリ
6-3
7-2
4-2
8-2
3-2
2-2
5-2
6-2
7-1
4-1
8-1
7-0
3-1
2-1
5-1
6-1
10
演算制御部
4-0
8-0
3-0
2-0
5-0
6-0
入力変数レジスタ
1

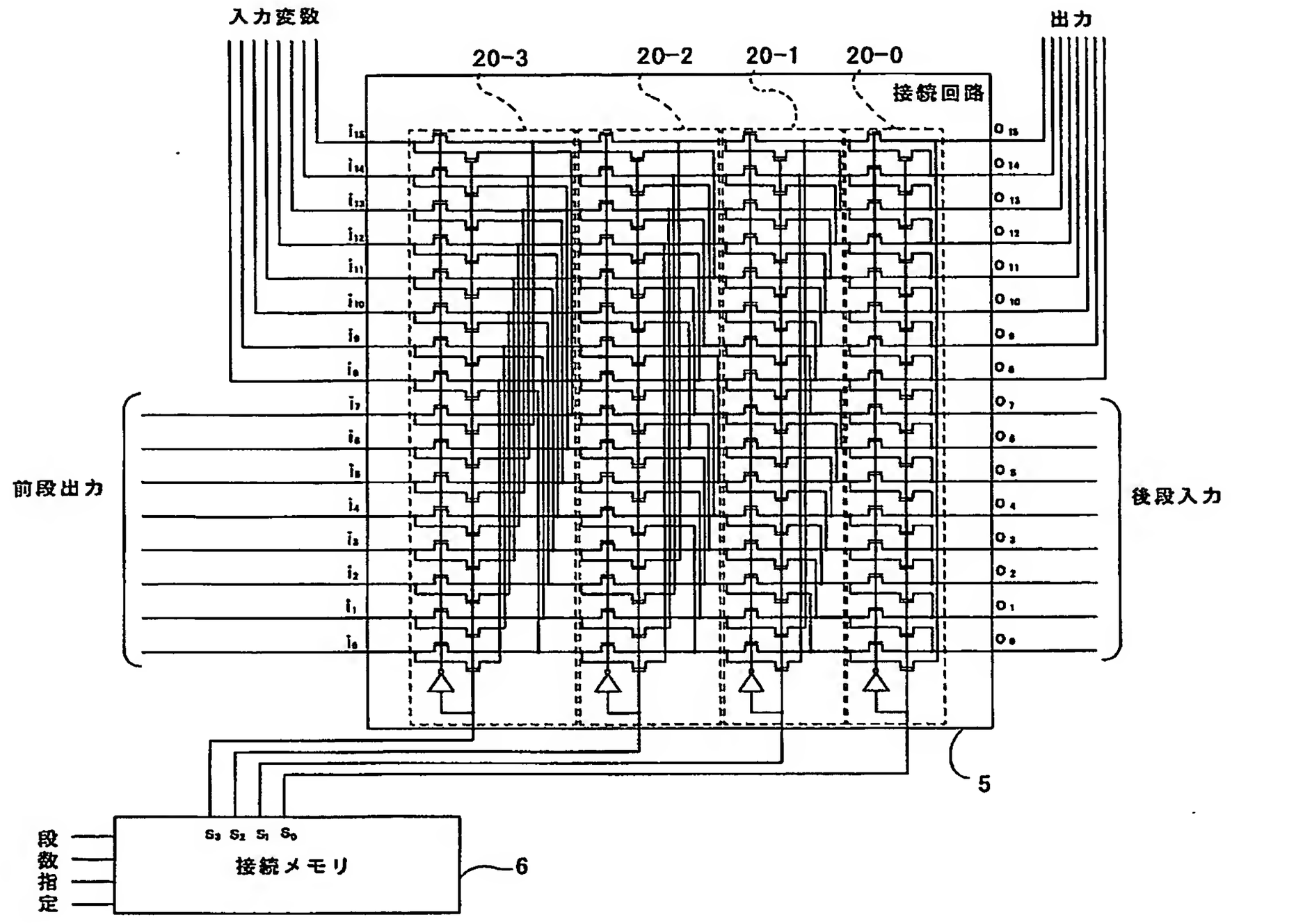
【図２】


11-3
11-2
11-1
11-0
2-i
in(16)
in(15)
in(14)
in(13)
in(12)
in(11)
in(10)
in(09)
in(08)
in(07)
in(06)
in(05)
in(04)
in(03)
in(02)
in(01)
in(00)
out(07)
out(06)
out(05)
out(04)
out(03)
out(02)
out(01)
out(00)
shf3
shf2
shf1
shf0

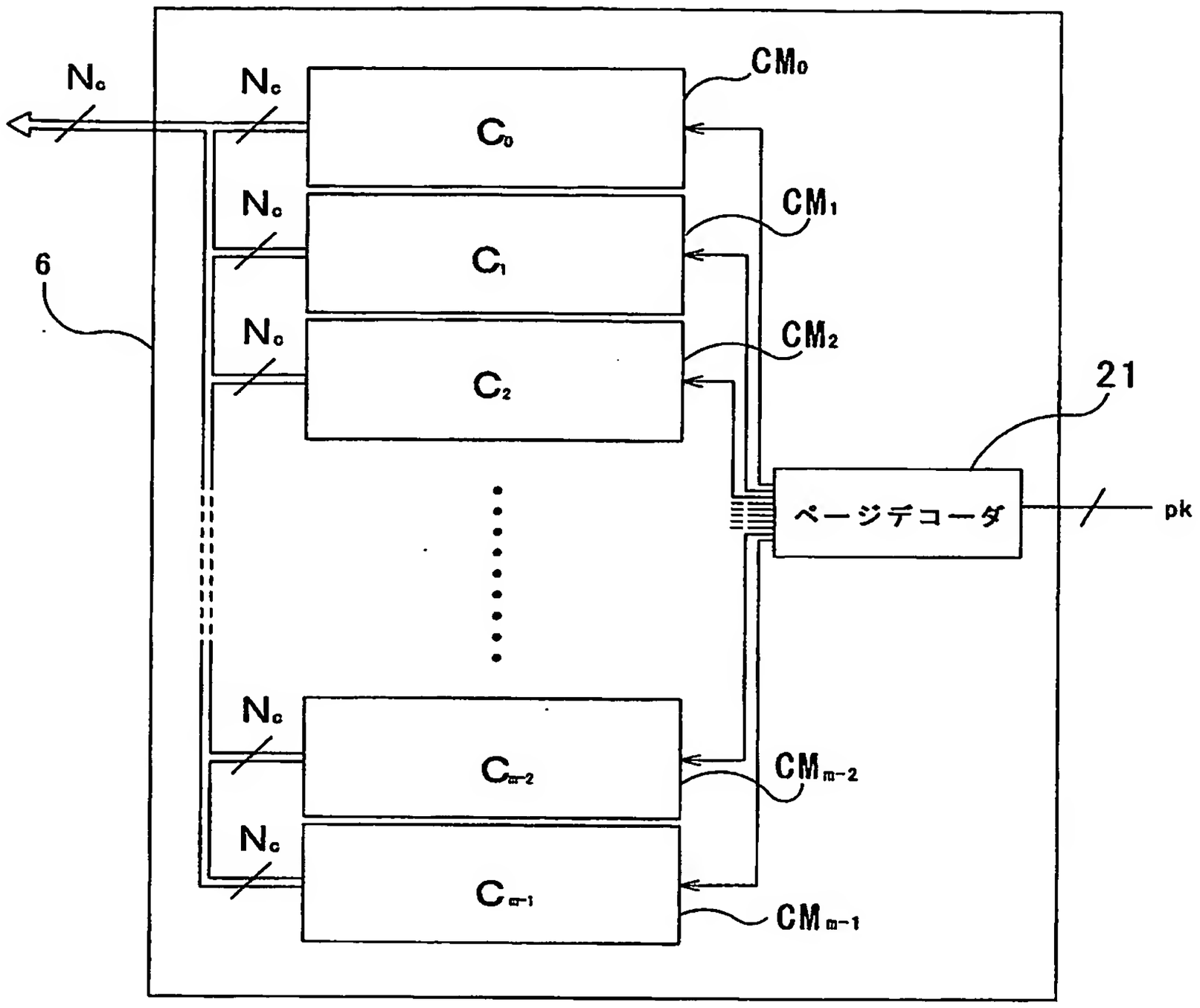
【図３】


15
4
p(pk)
ページ
デコーダ
FM0
Nout
Xk , Yk
Nin
アドレス
デコーダ
LUT(0)
Yk+1
17
18
FM1
16
LUT(1)
LUT(m-1)
FMm-1

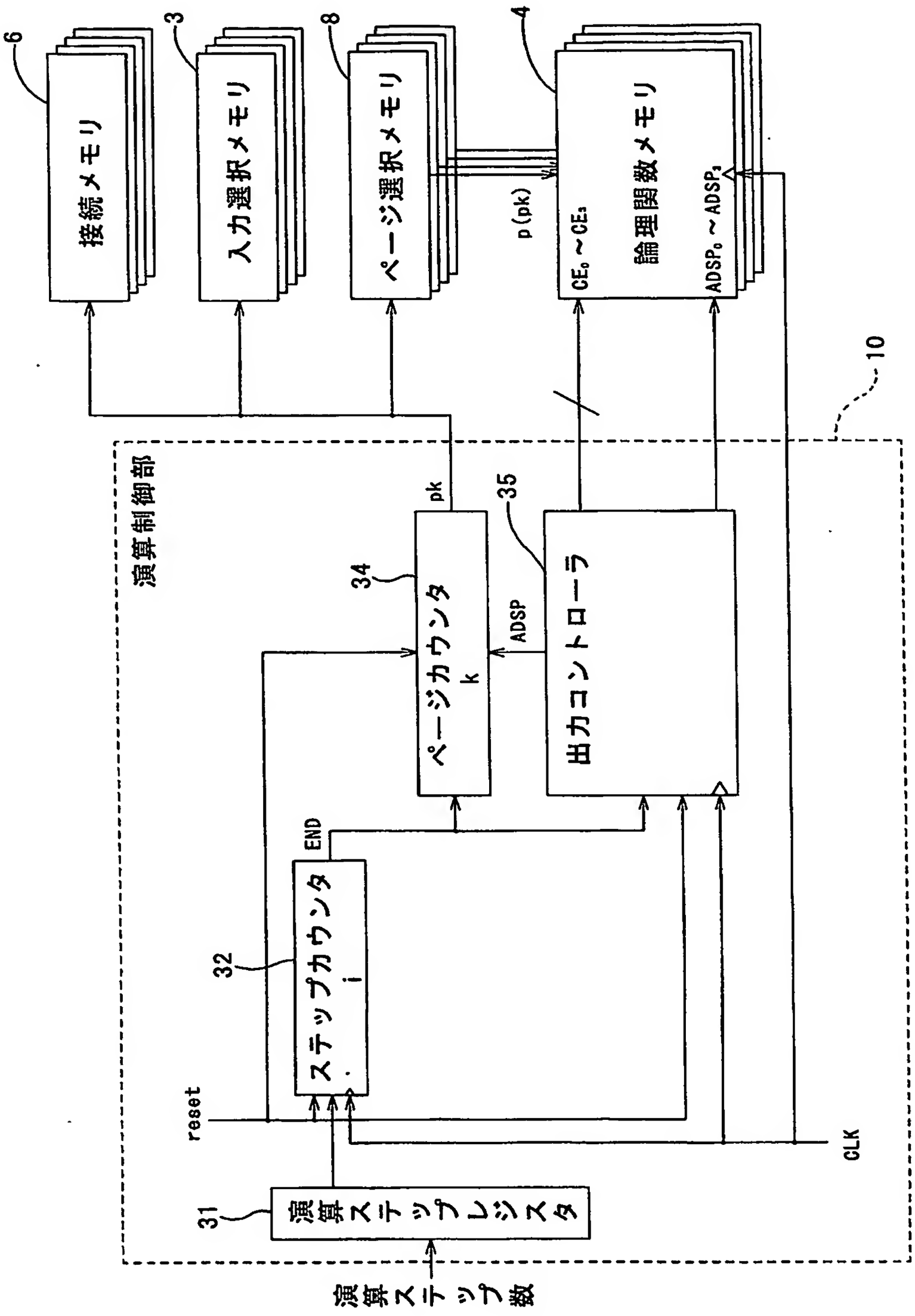
【図４】



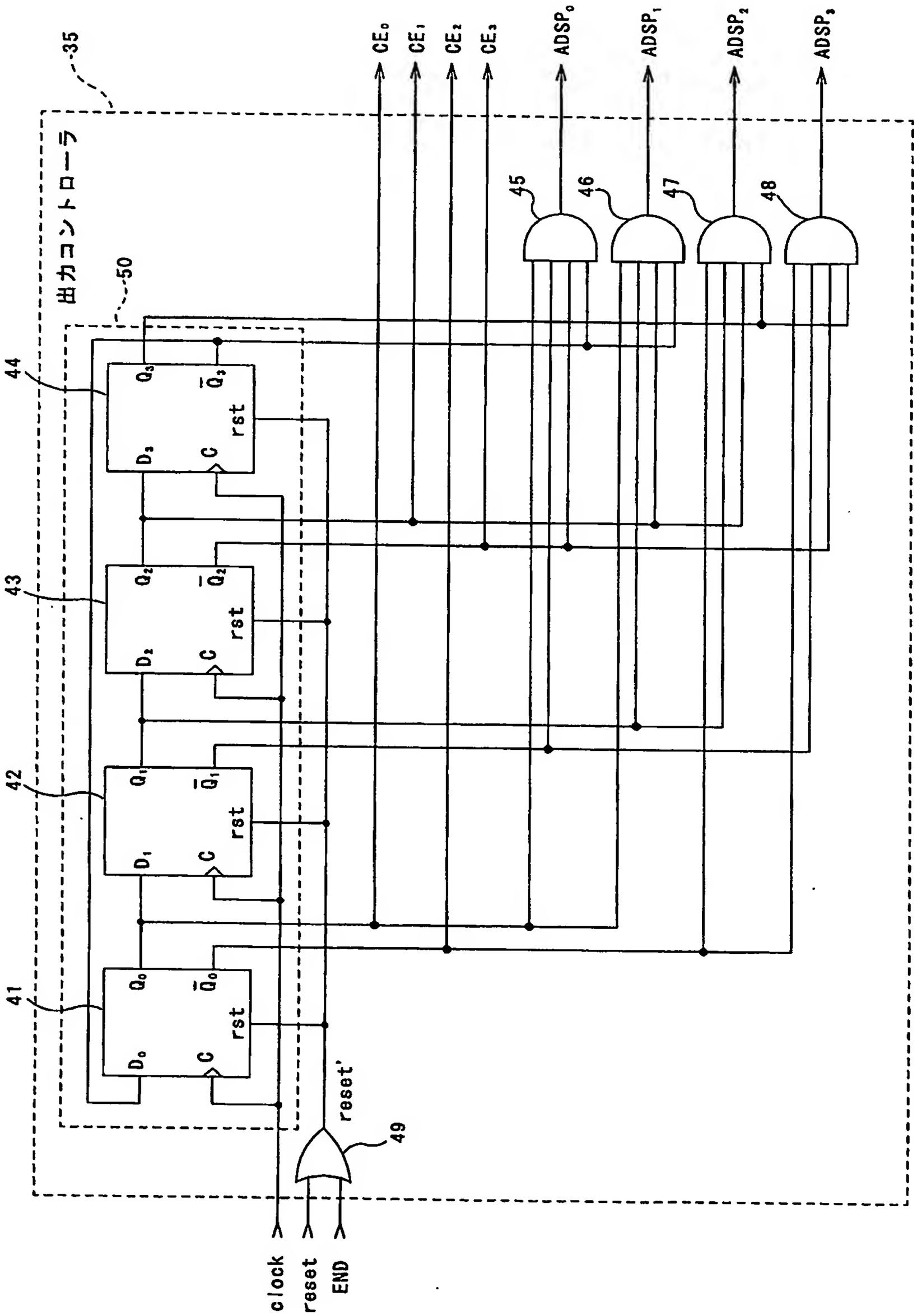
【図５】


Nc
Nc
CM0
C0
Nc
CM1
C1
6
Nc
CM2
C2
21
ページデコーダ
pk
Nc
Cm-2
CMm-2
Nc
Cm-1
CMm-1

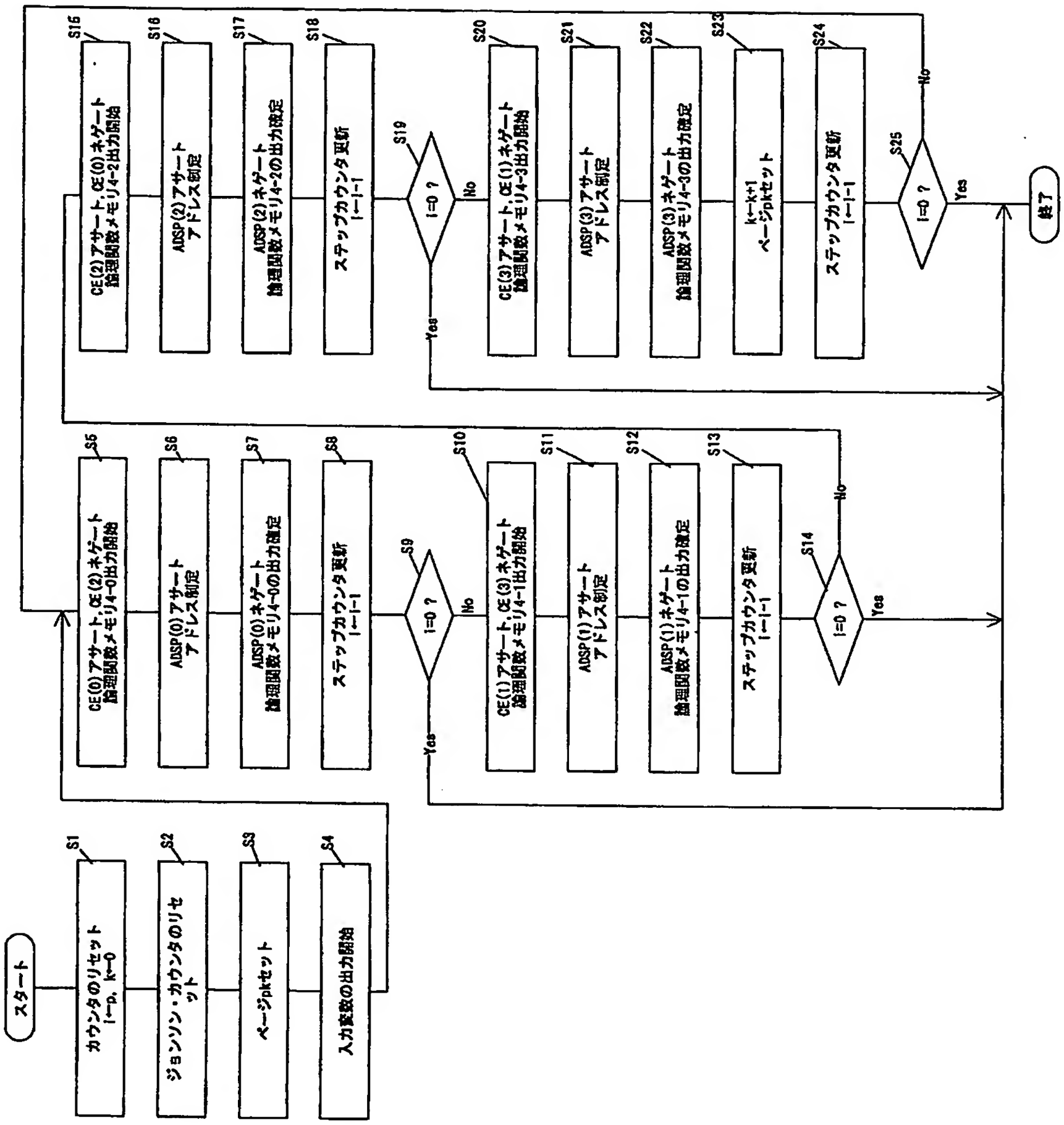
【図６】


6
接続メモリ
3
入力選択メモリ
8
ページ選択メモリ
4
論理関数メモリ
p(pk)
CE0～CE3
ADSP0～ADSP3
10
演算制御部
pk
34
ページカウンタ
k
ADSP
35
出力コントローラ
END
32
ステップカウンタ
i
reset
CLK
31
演算ステップレジスタ
演算ステップ数

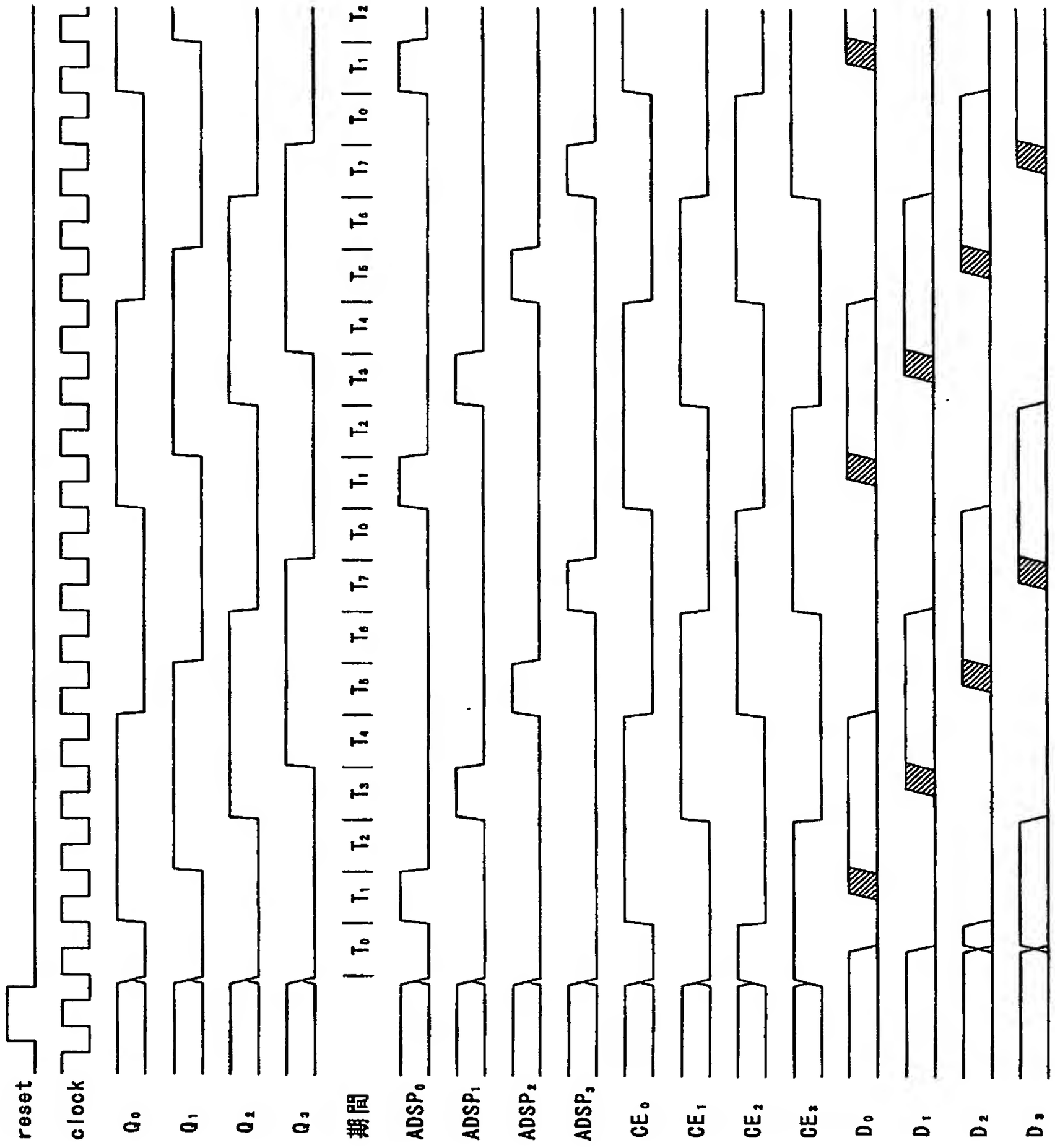
【図７】


35
出力コントローラ
50
CE0
CE1
CE2
CE3
ADSP0
ADSP1
ADSP2
ADSP3
45
46
47
48
44
43
42
41
Q3
Q̄3
D3
C
rst
Q2
Q̄2
D2
Q1
Q̄1
D1
Q0
Q̄0
D0
reset'
49
clock
reset
END

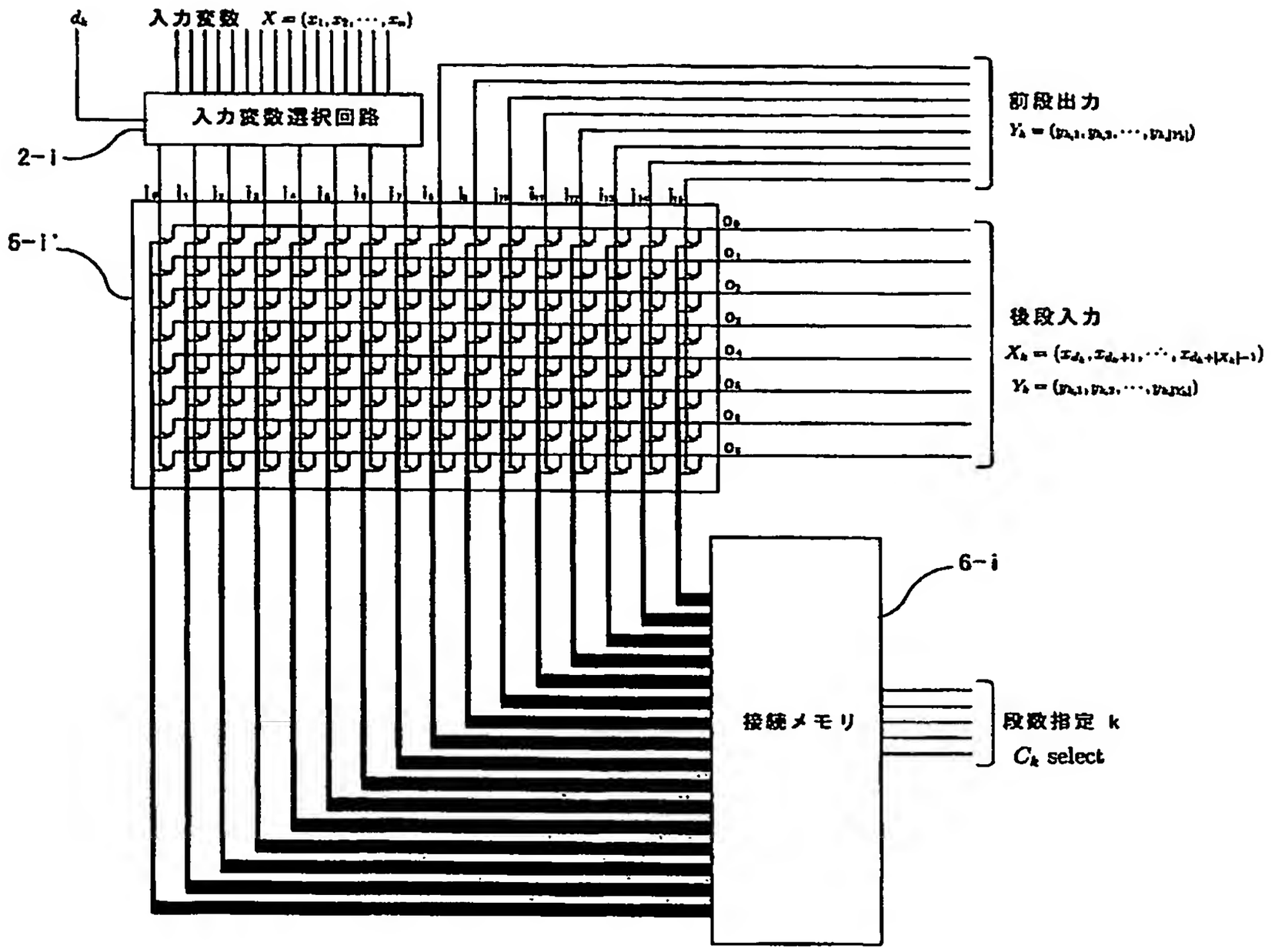
【図８】


スタート
S1
カウンタのリセット
i←p, k←0
S2
ジョンソン・カウンタのリセット
S3
ページpkセット
S4
入力変数の出力開始
S5
CE(0)アサート, CE(2)ネゲート
論理関数メモリ4-0出力開始
S6
ADSP(0)アサート
アドレス制定
S7
ADSP(0)ネゲート
論理関数メモリ4-0の出力確定
S8
ステップカウンタ更新
i←i-1
S9
i=0 ?
No
Yes
S10
CE(1)アサート, CE(3)ネゲート
論理関数メモリ4-1出力開始
S11
ADSP(1)アサート
アドレス制定
S12
ADSP(1)ネゲート
論理関数メモリ4-1の出力確定
S13
ステップカウンタ更新
i←i-1
S14
i=0 ?
No
Yes
S15
CE(2)アサート, CE(0)ネゲート
論理関数メモリ4-2出力開始
S16
ADSP(2)アサート
アドレス制定
S17
ADSP(2)ネゲート
論理関数メモリ4-2の出力確定
S18
ステップカウンタ更新
i←i-1
S19
i=0 ?
No
Yes
S20
CE(3)アサート, CE(1)ネゲート
論理関数メモリ4-3出力開始
S21
ADSP(3)アサート
アドレス制定
S22
ADSP(3)ネゲート
論理関数メモリ4-3の出力確定
S23
k←k+1
ページpkセット
S24
ステップカウンタ更新
i←i-1
S25
i=0 ?
No
Yes
終了

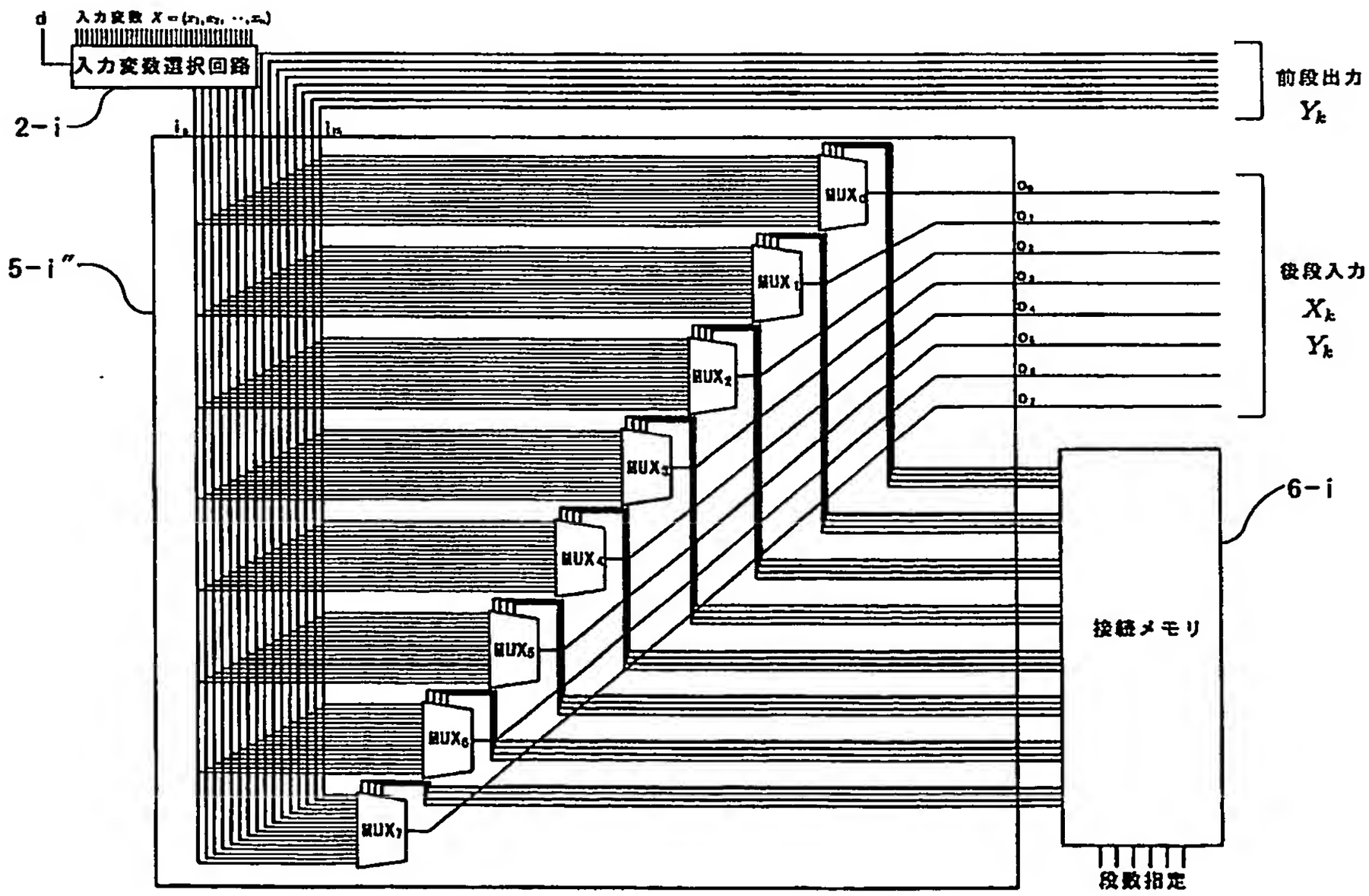
【図９】


reset
clock
Q0
Q1
Q2
Q3
期間
ADSP0
ADSP1
ADSP2
ADSP3
CE0
CE1
CE2
CE3
D0
D1
D2
D3
T0 T1 T2 T3 T4 T5 T6 T7 T0 T1 T2 T3 T4 T5 T6 T7 T0 T1 T2

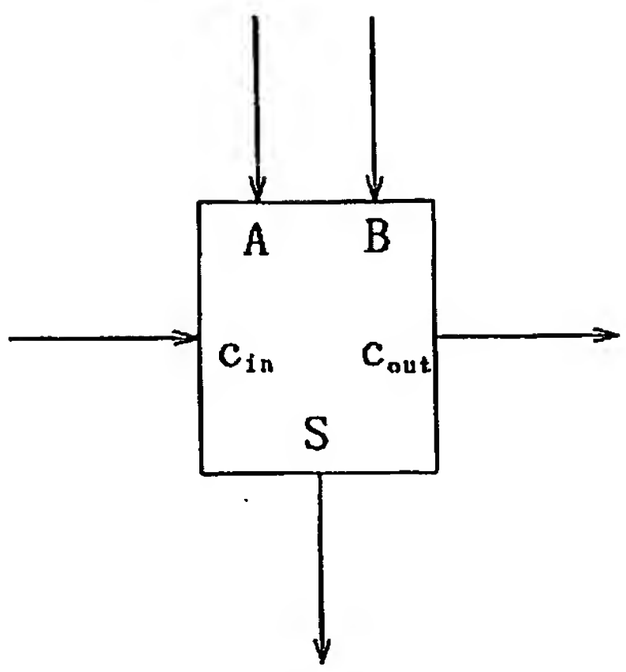
【図１０】


入力変数 $X=(x_1,x_2,\cdots,x_n)$
入力変数選択回路
2-i
5-i'
前段出力
$Y_k=(y_{k,1},y_{k,2},\cdots,y_{k,|Y_k|})$
後段入力
$X_k=(x_{d_k},x_{d_k+1},\cdots,x_{d_k+|X_k|-1})$
$Y_k=(y_{k,1},y_{k,2},\cdots,y_{k,|Y_k|})$
6-i
接続メモリ
段数指定 k
C_k select

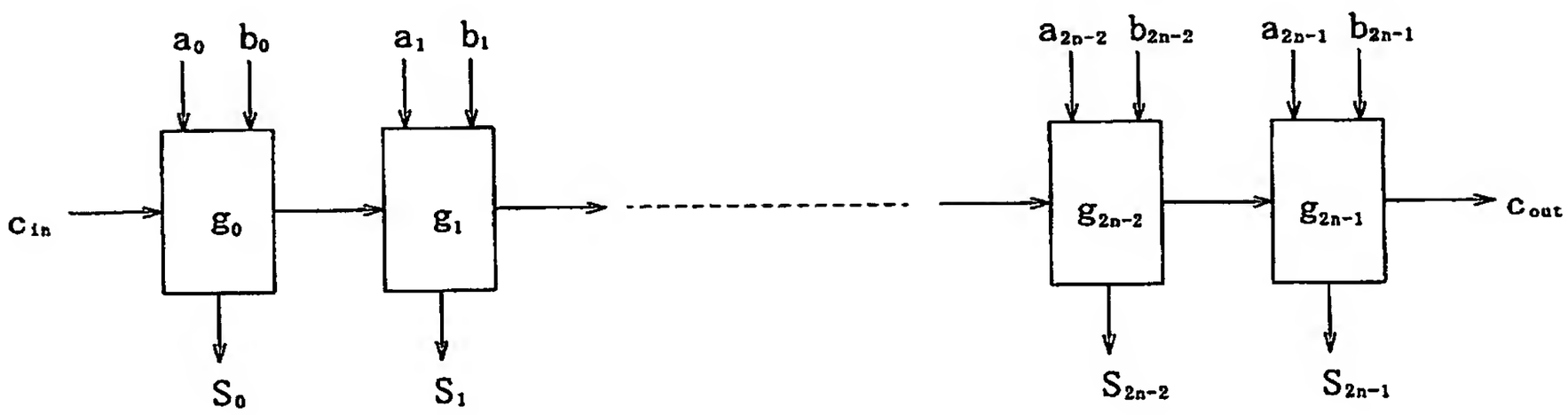
【図１１】


d
入力変数 $X=(x_1,x_2,\cdots,x_n)$
入力変数選択回路
2-i
5-i''
前段出力
Y_k
後段入力
X_k
Y_k
6-i
接続メモリ
段数指定
MUX0
MUX1
MUX2
MUX3
MUX4
MUX5
MUX6
MUX7

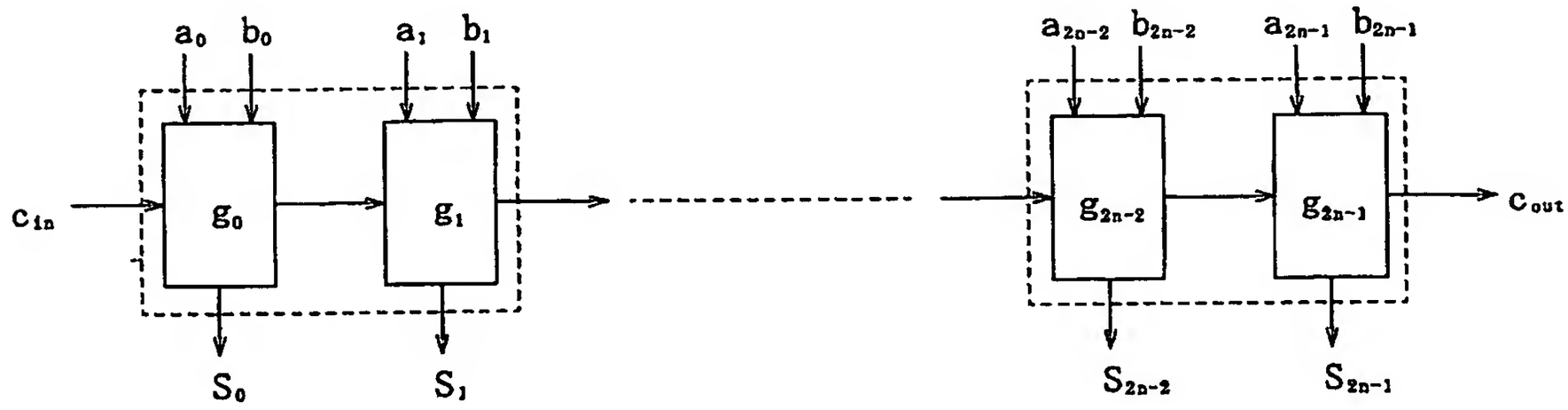
【図１２】


A
B
C_{in}
C_{out}
S

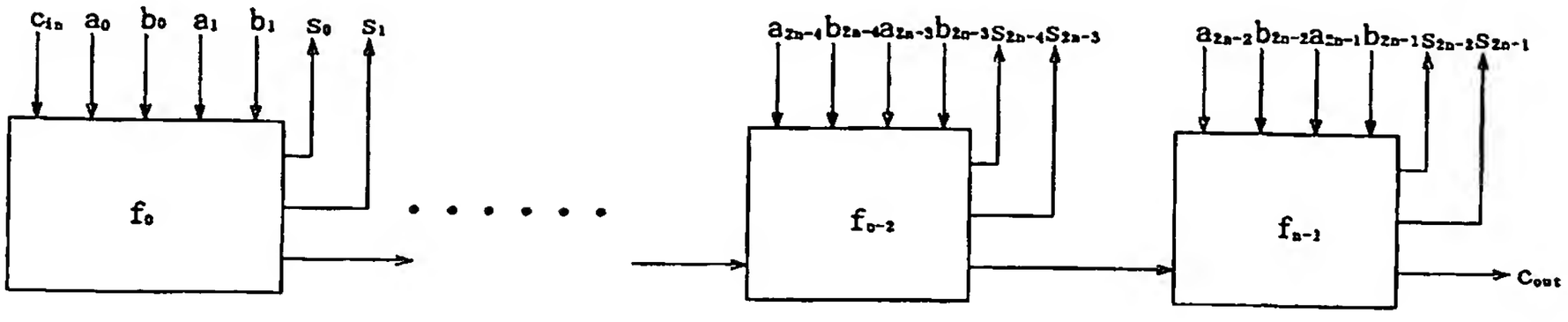
【図１３】


a_0 b_0
a_1 b_1
a_{2n-2} b_{2n-2}
a_{2n-1} b_{2n-1}
c_{in}
g_0
g_1
g_{2n-2}
g_{2n-1}
c_{out}
S_0
S_1
S_{2n-2}
S_{2n-1}

【図１４】


a_0 b_0
a_1 b_1
a_{2n-2} b_{2n-2}
a_{2n-1} b_{2n-1}
c_{in}
g_0
g_1
g_{2n-2}
g_{2n-1}
c_{out}
S_0
S_1
S_{2n-2}
S_{2n-1}

【図１５】


c_{in} a_0 b_0 a_1 b_1 s_0 s_1
f_0
a_{2n-4} b_{2n-4} a_{2n-3} b_{2n-3} S_{2n-4} S_{2n-3}
f_{n-2}
a_{2n-2} b_{2n-2} a_{2n-1} b_{2n-1} S_{2n-2} S_{2n-1}
f_{n-1}
c_{out}

【図１６】

(a)

c_{in}	a_1	a_0	b_1	b_0	c_{out}	s_1	s_0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	1
0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0

c_{in}	a_1	a_0	b_1	b_0	c_{out}	s_1	s_0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	0	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	1	0	0
1	1	1	0	1	1	0	1
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

(b)

アドレス								出力値							
i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
0	0	0	0	0	–	–	–	–	–	–	0	0	0	–	–
0	0	0	0	1	–	–	–	–	–	–	0	0	1	–	–
0	0	0	1	0	–	–	–	–	–	–	0	1	0	–	–
0	0	0	1	1	–	–	–	–	–	–	0	1	1	–	–
0	0	1	0	0	–	–	–	–	–	–	0	0	1	–	–
0	0	1	0	1	–	–	–	–	–	–	0	1	0	–	–
0	0	1	1	0	–	–	–	–	–	–	0	1	1	–	–
0	0	1	1	1	–	–	–	–	–	–	1	0	0	–	–
0	1	0	0	0	–	–	–	–	–	–	0	1	0	–	–
0	1	0	0	1	–	–	–	–	–	–	0	1	1	–	–
0	1	0	1	0	–	–	–	–	–	–	1	0	0	–	–
0	1	0	1	1	–	–	–	–	–	–	1	0	1	–	–
0	1	1	0	0	–	–	–	–	–	–	0	1	1	–	–
0	1	1	0	1	–	–	–	–	–	–	1	0	0	–	–
0	1	1	1	0	–	–	–	–	–	–	1	0	1	–	–
0	1	1	1	1	–	–	–	–	–	–	1	1	0	–	–

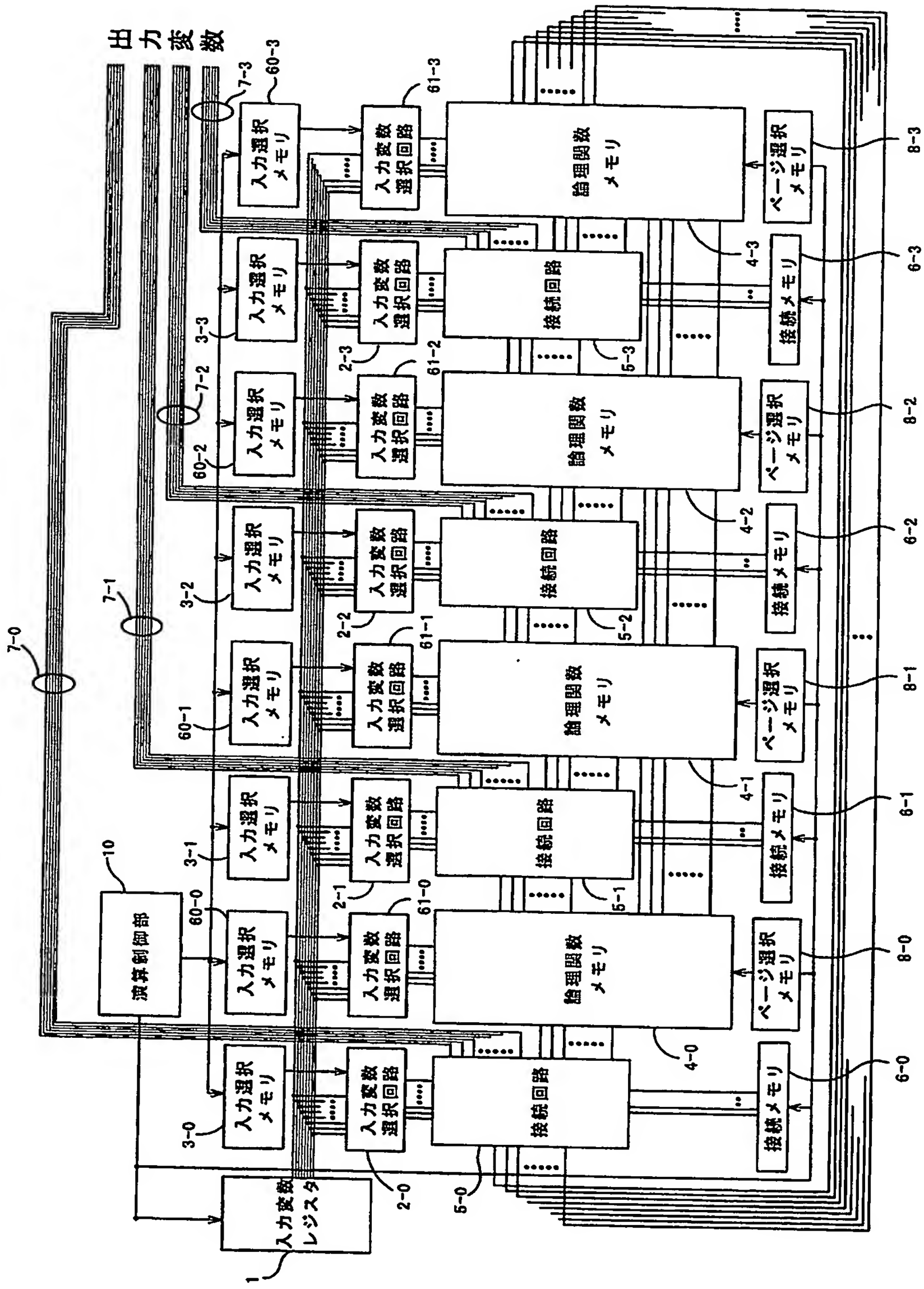
アドレス								出力値							
i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
1	0	0	0	0	–	–	–	–	–	–	0	0	1	–	–
1	0	0	0	1	–	–	–	–	–	–	0	1	0	–	–
1	0	0	1	0	–	–	–	–	–	–	0	1	1	–	–
1	0	0	1	1	–	–	–	–	–	–	1	0	0	–	–
1	0	1	0	0	–	–	–	–	–	–	0	1	0	–	–
1	0	1	0	1	–	–	–	–	–	–	0	1	1	–	–
1	0	1	1	0	–	–	–	–	–	–	1	0	0	–	–
1	0	1	1	1	–	–	–	–	–	–	1	0	1	–	–
1	1	0	0	0	–	–	–	–	–	–	0	1	1	–	–
1	1	0	0	1	–	–	–	–	–	–	1	0	0	–	–
1	1	0	1	0	–	–	–	–	–	–	1	0	1	–	–
1	1	0	1	1	–	–	–	–	–	–	1	1	0	–	–
1	1	1	0	0	–	–	–	–	–	–	1	0	0	–	–
1	1	1	0	1	–	–	–	–	–	–	1	0	1	–	–
1	1	1	1	0	–	–	–	–	–	–	1	1	0	–	–
1	1	1	1	1	–	–	–	–	–	–	1	1	1	–	–

(c)

アドレス								出力値							
i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
0	–	–	–	0	0	0	0	–	–	–	0	0	0	–	–
0	–	–	–	0	0	0	1	–	–	–	0	0	1	–	–
0	–	–	–	0	0	1	0	–	–	–	0	1	0	–	–
0	–	–	–	0	0	1	1	–	–	–	0	1	1	–	–
0	–	–	–	0	1	0	0	–	–	–	0	0	1	–	–
0	–	–	–	0	1	0	1	–	–	–	0	1	0	–	–
0	–	–	–	0	1	1	0	–	–	–	0	1	1	–	–
0	–	–	–	0	1	1	1	–	–	–	1	0	0	–	–
0	–	–	–	1	0	0	0	–	–	–	0	1	0	–	–
0	–	–	–	1	0	0	1	–	–	–	0	1	1	–	–
0	–	–	–	1	0	1	0	–	–	–	1	0	0	–	–
0	–	–	–	1	0	1	1	–	–	–	1	0	1	–	–
0	–	–	–	1	1	0	0	–	–	–	0	1	1	–	–
0	–	–	–	1	1	0	1	–	–	–	1	0	0	–	–
0	–	–	–	1	1	1	0	–	–	–	1	0	1	–	–
0	–	–	–	1	1	1	1	–	–	–	1	1	0	–	–

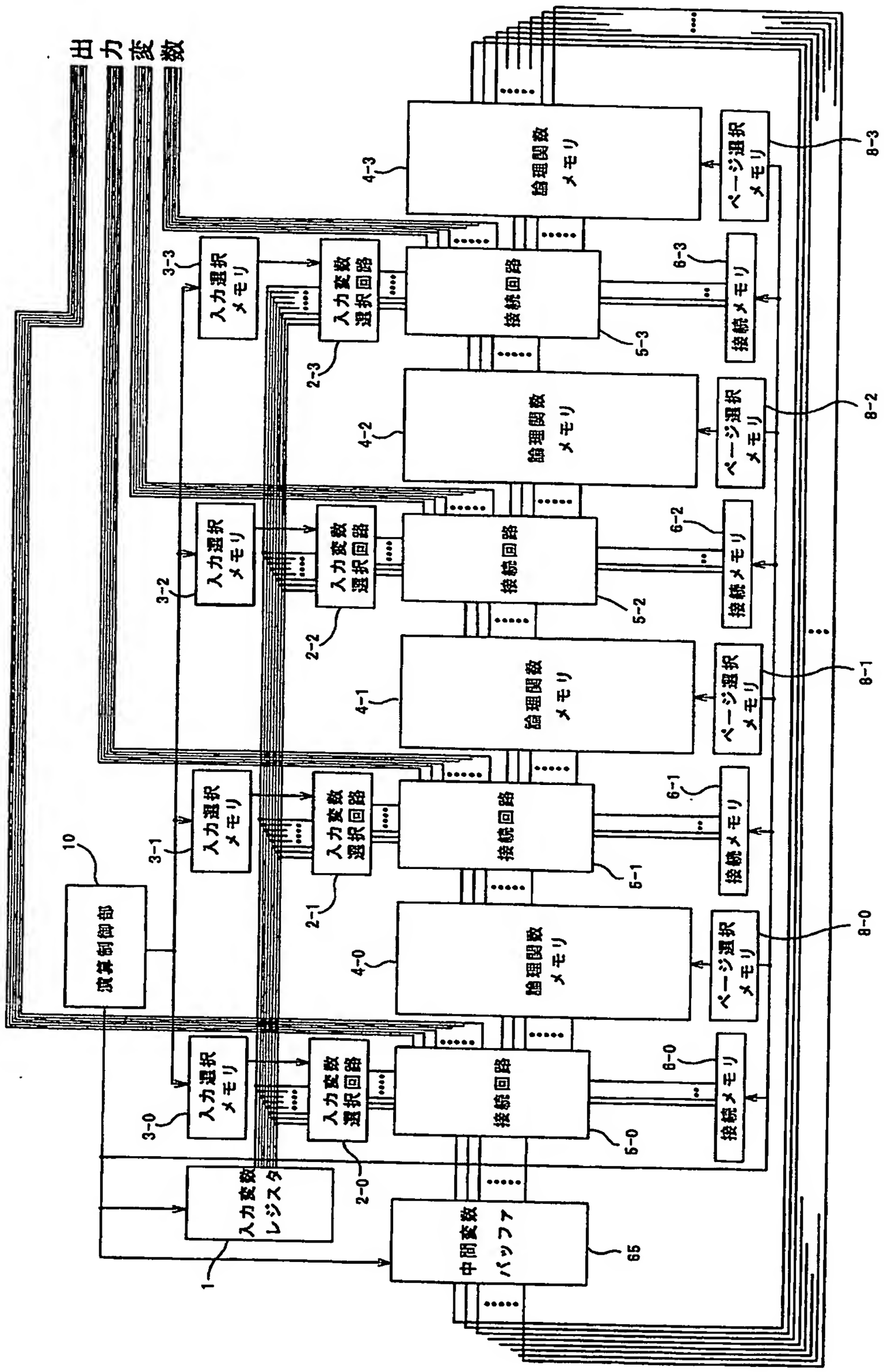
アドレス								出力値							
i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
1	–	–	–	0	0	0	0	–	–	–	0	0	1	–	–
1	–	–	–	0	0	0	1	–	–	–	0	1	0	–	–
1	–	–	–	0	0	1	0	–	–	–	0	1	1	–	–
1	–	–	–	0	0	1	1	–	–	–	1	0	0	–	–
1	–	–	–	0	1	0	0	–	–	–	0	1	0	–	–
1	–	–	–	0	1	0	1	–	–	–	0	1	1	–	–
1	–	–	–	0	1	1	0	–	–	–	1	0	0	–	–
1	–	–	–	0	1	1	1	–	–	–	1	0	1	–	–
1	–	–	–	1	0	0	0	–	–	–	0	1	1	–	–
1	–	–	–	1	0	0	1	–	–	–	1	0	0	–	–
1	–	–	–	1	0	1	0	–	–	–	1	0	1	–	–
1	–	–	–	1	0	1	1	–	–	–	1	1	0	–	–
1	–	–	–	1	1	0	0	–	–	–	1	0	0	–	–
1	–	–	–	1	1	0	1	–	–	–	1	0	1	–	–
1	–	–	–	1	1	1	0	–	–	–	1	1	0	–	–
1	–	–	–	1	1	1	1	–	–	–	1	1	1	–	–

【図１７】

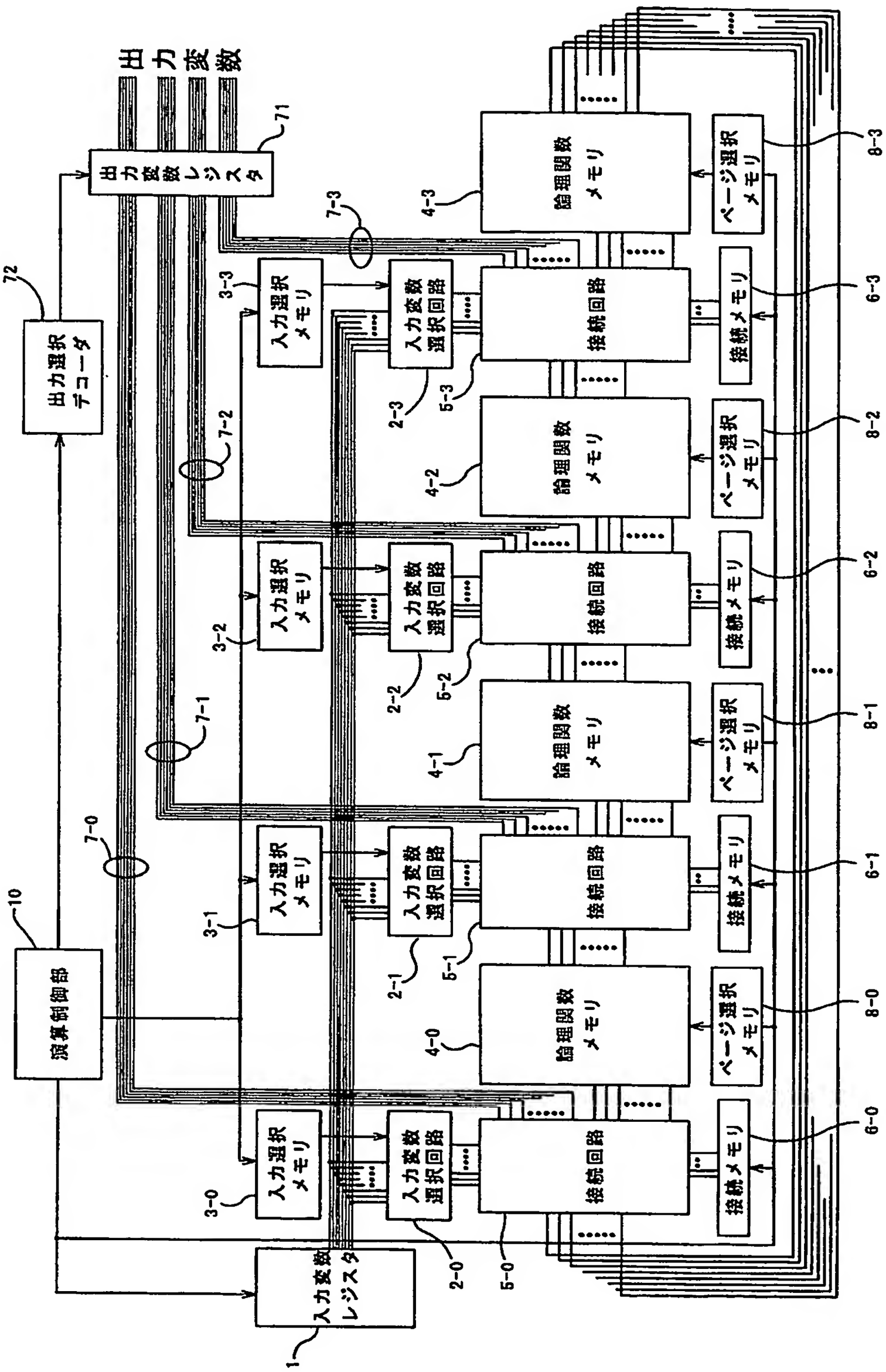


出証特２００４－３０４０１５０

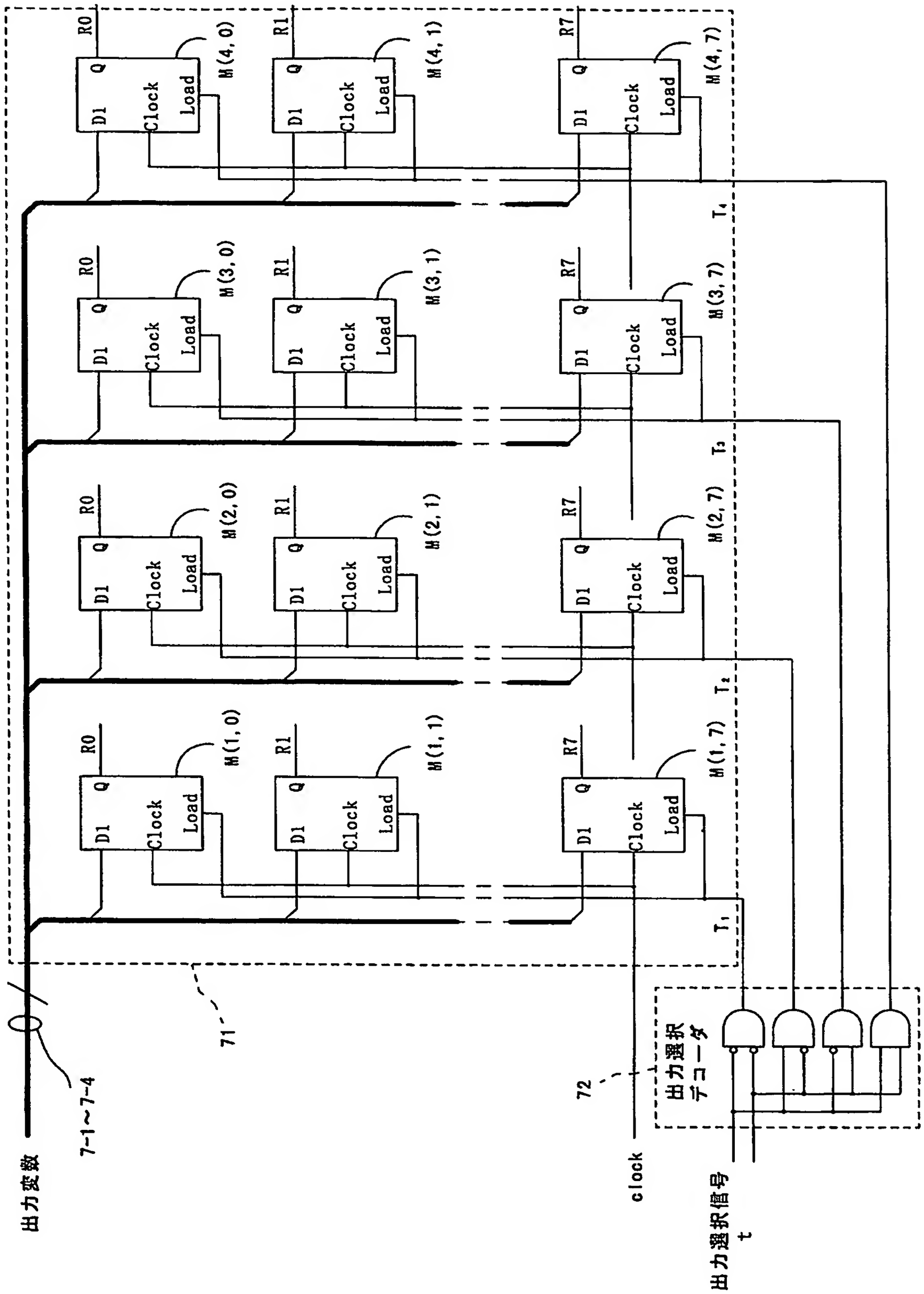
【図１８】


出力変数
論理関数メモリ
ページ選択メモリ
入力選択メモリ
入力変数選択回路
接続回路
接続メモリ
演算制御部
入力変数レジスタ
中間変数バッファ
4-3
8-3
3-3
2-3
5-3
6-3
4-2
8-2
3-2
2-2
5-2
6-2
4-1
8-1
3-1
2-1
5-1
6-1
10
4-0
8-0
3-0
2-0
5-0
6-0
1
65

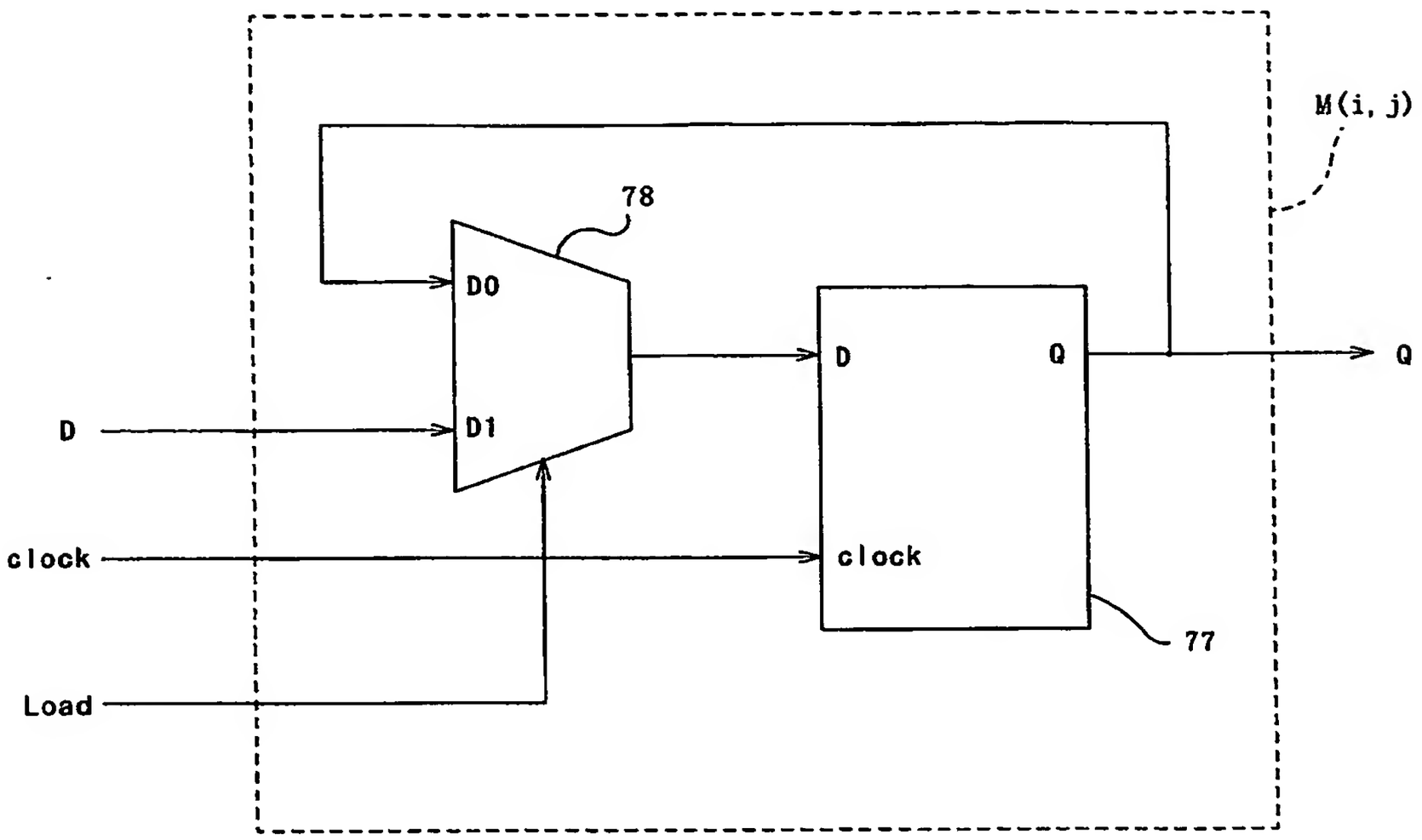
【図１９】


出力変数
出力変数レジスタ
71
72
出力選択デコーダ
10
演算制御部
7-0
7-1
7-2
7-3
入力変数レジスタ
1
入力選択メモリ
3-0
3-1
3-2
3-3
入力変数選択回路
2-0
2-1
2-2
2-3
接続回路
5-0
5-1
5-2
5-3
接続メモリ
6-0
6-1
6-2
6-3
論理関数メモリ
4-0
4-1
4-2
4-3
ページ選択メモリ
8-0
8-1
8-2
8-3

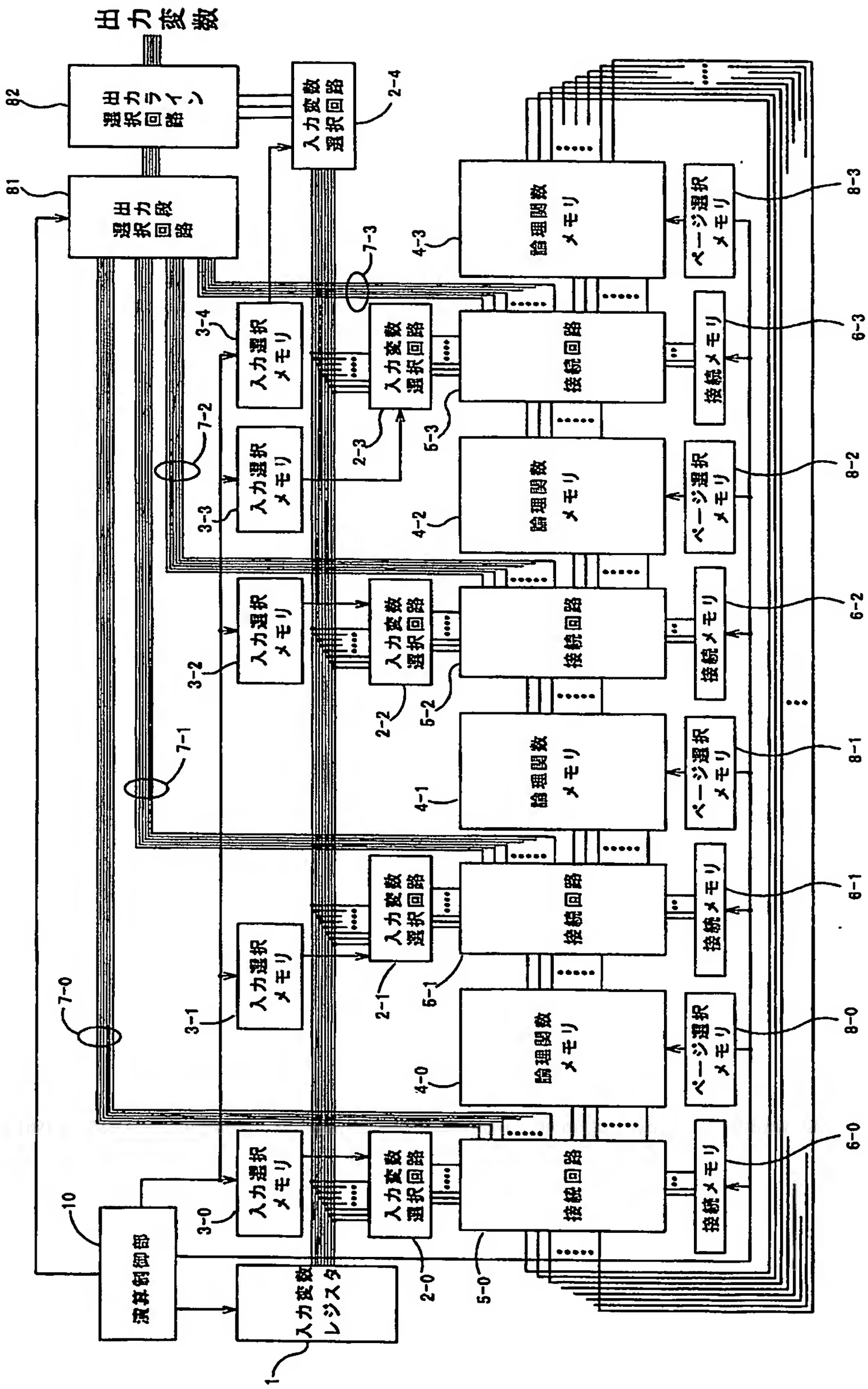
【図２０】


R0
R1
R7
Q
D1
Clock
Load
M(4,0)
M(4,1)
M(4,7)
M(3,0)
M(3,1)
M(3,7)
M(2,0)
M(2,1)
M(2,7)
M(1,0)
M(1,1)
M(1,7)
T4
T3
T2
T1
71
72
7-1～7-4
出力変数
clock
出力選択
デコーダ
出力選択信号
t

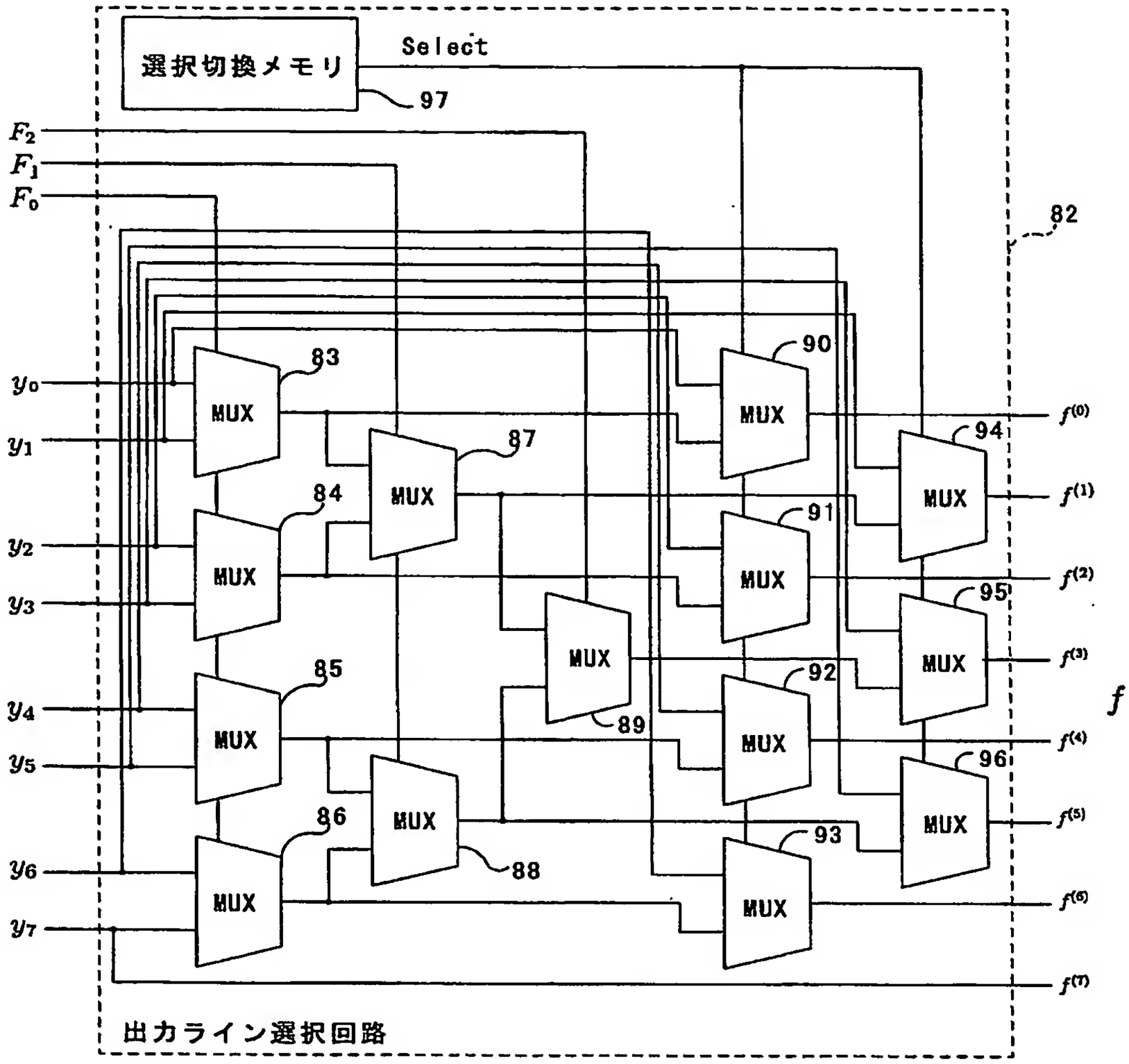
【図２１】



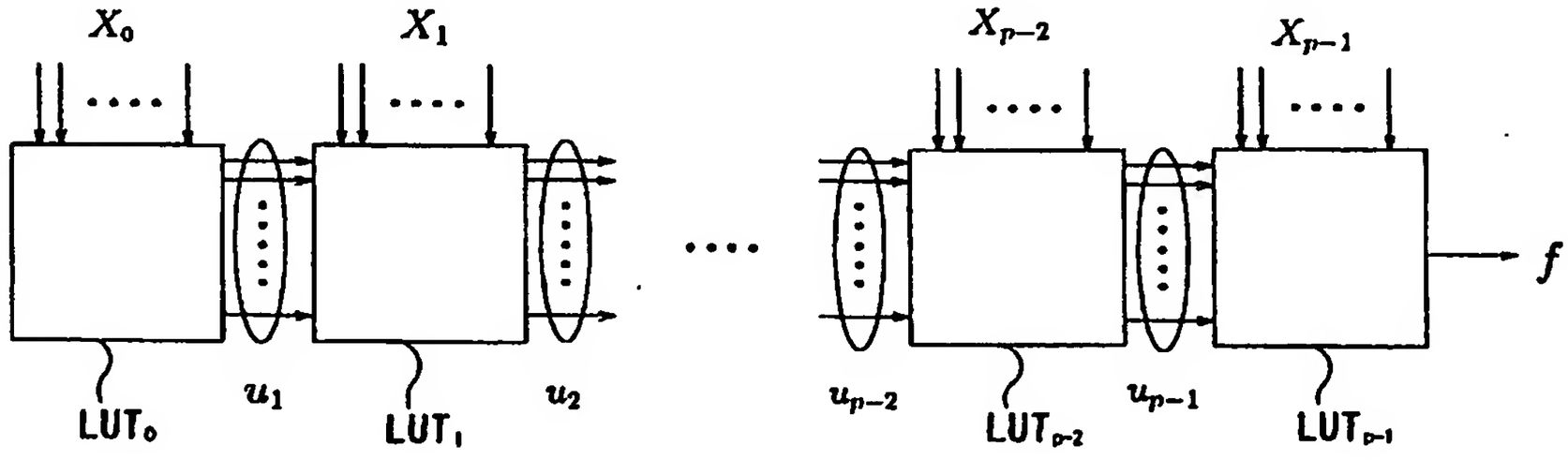
【図２２】


出力変数
選択出力ライン
選択出力回路
入力変数選択回路
選択出力段回路
入力選択メモリ
入力変数選択回路
論理関数メモリ
ページ選択メモリ
接続回路
接続メモリ
演算制御部
入力変数レジスタ
7-0
7-1
7-2
7-3
8-0
8-1
8-2
8-3
6-0
6-1
6-2
6-3
5-0
5-1
5-2
5-3
4-0
4-1
4-2
4-3
3-0
3-1
3-2
3-3
3-4
2-0
2-1
2-2
2-3
2-4
1
10
81
82

【図２３】


選択切換メモリ
Select
97
F_2
F_1
F_0
82
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
MUX
83
84
85
86
87
88
89
90
91
92
93
94
95
96
$f^{(0)}$
$f^{(1)}$
$f^{(2)}$
$f^{(3)}$
$f^{(4)}$
$f^{(5)}$
$f^{(6)}$
$f^{(7)}$
f
出力ライン選択回路

【図２４】


X_0
X_1
X_{p-2}
X_{p-1}
u_1
u_2
u_{p-2}
u_{p-1}
LUT_0
LUT_1
LUT_{p-2}
LUT_{p-1}
f

【書類名】　要約書

【要約】

【課題】　目的論理関数に応じ入力変数・レイル数を柔軟に変化でき、分解関数の数に応じてカスケード段数を柔軟に変化できるルックアップテーブル・カスケード論理回路を提供する。

【解決手段】　各分解関数をルックアップテーブルとして記憶する、直列に配列された複数の論理関数メモリ４－０～４－３、前段の論理関数メモリの出力のうち、後段の論理関数メモリの入力に接続されるものの接続情報を記憶する接続メモリ６－０～６－３、接続メモリより出力される接続順序に従い、前段の論理関数メモリの出力と後段の論理関数メモリの入力を選択的に接続する接続回路５－１～５－３、及び最後段の論理関数メモリの出力を最前段の論理関数メモリに選択的に入力するフィードバック回路５－０，６－０を備えた構成とした。

【選択図】　図１

認定・付加情報

特許出願の番号	特願２００３－１０５７６２
受付番号	５０３００５９０２９０
書類名	特許願
担当官	第八担当上席　　００９７
作成日	平成１５年　４月１０日

＜認定情報・付加情報＞

【提出日】　平成15年 4月 9日

次頁無

【書類名】　出願人名義変更届
【あて先】　特許庁長官殿
【事件の表示】
　【出願番号】　特願2003-105762
【承継人】
　【識別番号】　802000031
　【氏名又は名称】　財団法人北九州産業学術推進機構
　【代表者】　有馬　朗人
【承継人代理人】
　【識別番号】　100121371
　【弁理士】
　【氏名又は名称】　石田　和人
　【電話番号】　093-695-3113
【手数料の表示】
　【予納台帳番号】　222495
　【納付金額】　4,200円
【提出物件の目録】
　【物件名】　承継人であることを証する書面　1
　　【援用の表示】　平成１５年１２月１２日提出の手続補足書により補足する。
　【物件名】　委任状　1
　　【援用の表示】　平成１５年１２月１２日提出のものを援用する。

認定・付加情報

特許出願の番号	特願２００３－１０５７６２
受付番号	５０３０２０５３２９３
書類名	出願人名義変更届
担当官	古田島　千恵子　　７２８８
作成日	平成１６年　３月２５日

＜認定情報・付加情報＞

【提出日】	平成15年12月12日
【承継人】	
【識別番号】	802000031
【住所又は居所】	福岡県北九州市若松区ひびきの２番１号
【氏名又は名称】	財団法人北九州産業学術推進機構
【承継人代理人】	申請人
【識別番号】	100121371
【住所又は居所】	福岡県北九州市若松区ひびきの２番１号　北九州学術研究都市産学連携センターＴ－３０２　石田特許事務所内
【氏名又は名称】	石田　和人

特願２００３－１０５７６２

出　願　人　履　歴　情　報

識別番号　　［５０２４２８４６７］

１．変更年月日　　２００２年１１月２６日
［変更理由］　　新規登録
住　所　　福岡県福岡市中央区今川１丁目１０－４３－４０３
氏　名　　笹尾　勤

２．変更年月日　　２００３年　５月２５日
［変更理由］　　住所変更
住　所　　福岡県福岡市中央区鳥飼３丁目１５－１７
氏　名　　笹尾　勤

特願２００３－１０５７６２

出願人履歴情報

識別番号　［８０２０００００３１］

１．変更年月日　２００２年　４月１９日
［変更理由］　新規登録
住　所　福岡県北九州市若松区ひびきの２番１号
氏　名　財団法人北九州産業学術推進機構

出証番号　出証特２００４－３０４０１５０